RL-TR-97-13
Final Technical Report
June 1997

ERROR DETECTION AND CORRECTION FOR OPTICAL MEMORIES

Colorado State University

Pericles A. Mitkas

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Rome, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-97-13 has been reviewed and is approved for publication.

Gary T Sunada

APPROVED:
GARY T. SUNADA, 2LT, USAF
Project Engineer

FOR THE COMMANDER:
JOSEPH CAMERA, Technical Director
Intelligence & Reconnaissance Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL/IRAP, 32 Hangar Road, Rome, NY 13441-4114. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE June 1997	3. REPORT TYPE AND DATES COVERED Final May 95 - May 96

4. TITLE AND SUBTITLE

ERROR DETECTION AND CORRECTION FOR OPTICAL MEMORIES

5. FUNDING NUMBERS

C - F30602-95-C-0067
PE - 62702F
PR - 4594
TA - 15
WU - P2

6. AUTHOR(S)

Pericles A. Mitkas

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Colorado State University
601 S. Howes Street
Fort Collins, CO 80523

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Rome Laboratory/IRAP
32 Hangar Road
Rome, NY 13441-4114

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

RL-TR-97-13

11. SUPPLEMENTARY NOTES

Rome Laboratory Project Engineer: Lt Gary T. Sunada/IRAP/(315) 330-4581

12a. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Novel optical data storage technologies, such as holographic memories, can store and retrieve data in the form of two-dimensional arrays of bits of pages (page-oriented memories, POMs). It is anticipated that such memories will soon be able to provide the data rates required by high performance systems. However, for this to be possible, data pages must be retrieved free of errors. Because there are several sources of noise in a POM system, the data stored in the memory must be appropriately encoded so that an error correction scheme will be able to remove the errors during retrieval and provide the required bit-error rate.

In this effort, we identified and classified several sources of errors in POMs. We developed a set of metrics for the evaluation of error correcting codes for POMs and we evaluated several array codes. We developed a simulator, where we incorporated implementations of most of the error generation processes and certain error correction codes. We used the simulator to evaluate and compare different error correction codes and to approximate real POM systems. Finally, we investigated optoelectronic techniques for the parallel implementation of error control schemes in POMs that could achieve the required throughput.

14. SUBJECT TERMS

Optical Memories, Page-Oriented Memories, Error Detection, Error Correction, Bit Error Rate, Error Simulation

15. NUMBER OF PAGES

68

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

TABLE OF CONTENTS

0. Statement of the Problem

In recent years, novel optical data storage technologies have shown significant progress and are soon expected to enter the commercial product stage. Such promising technologies include plane and volume holograms, two-photon three-dimensional memories, and spectral hole-burning storage. With the demonstration of high volumetric capacity and the potential for very high data rates, parallel optical memories can offer solutions to the secondary storage requirements of many input/output intensive computer applications, one of which is the management of very large databases. Most of these emerging technologies can store and retrieve data in the form of two-dimensional bit arrays or pages and thus, we are going to refer to them as *page-oriented memories* (POMs). This type of data access is radically different from the memory output format of conventional storage devices such as magnetic or optical disks and magnetic or optical tape. Over the next five years, it is anticipated that data rates in the range of 50 to 100 MB/sec will be required for high performance systems. Bit-serial output is generated by single disk drives, while several bits (one word) at a time can be read by storage devices with multiple heads and/or ports. The optical tape drive developed by CREO is one example of memory systems in the second category. In order to ensure bit error rates (BER) lower than 10^{-12} for binary data, commercial storage devices employ a variety of data encoding techniques which improve the raw bit error rates by several orders of magnitude. Error detection and correction is performed on-the-fly without slowing the data rate to the host.

To take full advantage of the high output rate that POMs can offer, any bottleneck between the memory and the host must be eliminated. Traditional error control methods have been designed to look at only a few bits at a time and they would be inefficient and slow for memories that can retrieve pages as large as 10^6 bits. Few measurements have been done to determine the raw BER in 3D optical memories and they report it in the range of $10^{-3} - 10^{-6}$. Though sufficient for images, such error rates are totally unacceptable for alphanumeric data. New encoding schemes are required to allow for fast, efficient, and parallel error control. The ideal scheme should be able to receive one page from the memory, perform the error control in a single step, and pass the corrected array (data bits only) to the host computer in parallel.

1. Objectives and Major Tasks

The main objective of this effort is to investigate the error control process during the data retrieval phase in optical memories that can generate a two-dimensional parallel output. Specific objectives include: a) the identification and classification of types of errors during optical readout, b) development and evaluation of encoding schemes for efficient error detection and correction, and c) investigation of optical/optoelectronic implementation of the error control process.

The major tasks as listed in the Statement Of Work (Sections 4.1.1–4.1.9) were as follows:

1.1 Classification of types of errors in page oriented memories (POMs),

1.2 Development of a set of metrics for evaluation of error control schemes for POMs,

1.3 Development of a software package for simulation and evaluation of the error control process,

1.4 Evaluation of existing array coding schemes for use in page oriented memories,

1.5 Development of encoding schemes for page oriented memories,

1.6 Investigation of optoelectronic techniques for parallel implementation of error detection and correction in POMs,

1.7 Continuous determination of the status of the effort through regular progress reports,

1.8 Final report,

1.9 Oral presentations at such times and places as designated by the Rome Laboratory personnel.

Tasks 1.1, 1.2, 1.3, and 1.7–1.9 were to be 100% complete by the end of this project. Research leading to the objectives associated with tasks 1.4–1.6 was to begin during this research period but was expected to continue for another two years with additional funding, if available.

2. Period Covered

The research covered in this report was performed at Colorado State University from May 10, 1995 to September 30, 1996.

3. Personnel

The research was performed by the PI, Dr. Pericles A. Mitkas, and his team that included Post Doctoral Fellow, Dr. George Betzos, and Graduate Research Assistants (GRAs), John Hutton, Michael Porter, and Maureen Schaffer. Graduate student Umesh Mehta also contributed to this project. Data from our volume holographic memory system was provided by GRA Keith Richling. GRA Alan Simone provided some image processing tools that were used to develop certain data recovery procedures in the simulator.

4. Accomplishments

We embarked on a systematic study of the error detection and correction process for parallel optical memories with two-dimensional output and established the framework that will enable the development and evaluation of appropriate encoding/decoding techniques. During the course of this effort, we assumed a simple generic model of a POM that can output a page of strictly binary data encoded in intensity or amplitude. Later, this model may be customized to better approximate the conditions for different types of memories and different types of data. We have already performed a more detailed study of the volume holographic memory technology.

In the following sections we describe our technical accomplishments with respect to the major tasks outlined in Section 1. Before we proceed, we must mention two additional tasks that were not listed in the original Statement Of Work:

a) The first W*orkshop on Data Encoding for Page-oriented Optical Memories (DEPOM'96)* was organized by the PI and his research team and held in Phoenix, AZ, March 27–28, 1996. The Workshop was sponsored by the Rome Laboratory, the Air Force Office of Scientific Research and Colorado State University. The purpose of this Workshop was to bring together researchers who are experts and/or have an interest in the following areas: Page-oriented optical memory (POM) systems; Array codes for error control in POMs; Error generating processes in POMs; Channel characterization and modelling; Data encoding schemes for POMs; Hardware schemes for parallel error control and decoding in POMs. DEPOM'96 was attended by 53 people representing 14 Universities, 10 companies, and 3 Government Agencies. The Workshop Proceedings will soon be published in a volume.

b) Every effort was made to obtain raw data from the output of various POM systems that are under development at several research institutions in the country. We were able to get data from

Demetri Psaltis' group at Caltech and used them in our simulator for testing and calibrating our data recovery techniques.

4.1 Classification of types of errors in page-oriented optical memories

We performed an exhaustive review of the literature which yielded a large number of publications. The most useful ones are listed in the Bibliography Section at the end of this report. Our investigation yielded a large number of possible error sources that were classified as in Table 1. For most of these errors, mathematical models were developed or identified in the literature and were programmed into the simulator.

Table 1. Classification of Error Sources in POMs

Errors due to the Nature of Data

1. Data dependent non-uniformity
2. Inter-symbol Interference

Errors in the Input Process

3. SLM contrast ratio
4. SLM pixel crosstalk
5. Fixed pattern on the SLM, (i.e., due to defective pixels)
6. Gaussian beam profile

Errors in the System

7. Alignment
8. Aberrations of the optical system
9. Optical defocusing
10. Dust
11. Magnification
12. Light scattering and reflection

Errors due to the Recording Material

13. Diffraction efficiency of the hologram
14. Diffraction effects
15. Recording media defects
16. Interpixel crosstalk or intersymbol interference
17. Edge blurring. (High frequency cut-off due to crystal size)
18. Interpage crosstalk (Fourier Hologram)
19. Fanning noise

Errors in the Output Process

20. CCD reset noise
21. CCD preamp noise
22. CCD integration time

23. Fixed pattern on the CCD (i.e., due to defective pixels)

4.2 Development of a set of metrics for evaluation of error control codes for POMs

The table below shows the set of parameters that we have selected to evaluate array codes.

Length of code:	n
Number of information bits:	k
Rate:	$r = \frac{k}{n}$
Coding overhead:	$n-k$
Coding gain:	$g = 10\log_{10}\frac{SNR}{SNR'}$ SNR : uncoded, SNR' : coded
Maximum number of correctable errors:	d
Maximum number of detectable errors:	t
Types of errors that can be corrected:	(random, burst, etc.)
Coding/decoding delay:	T_c, T_d
Hardware complexity:	H (i.e., number of gates)
Interconnection complexity:	I (i.e. number and topology of interconnects)
Coding/decoding algorithm complexity:	$O(\,)$
Parallel implementation possibility:	yes/no; if yes, degree of parallelism
Multiple page readout requirement:	yes/no; if yes, how many
Corrected bit error rate:	P_d, as a function of the raw bit error rate, P_e, of the system

The majority of these parameters can be represented in a single figure of merit (FOM) for the objective comparison of the array codes we evaluate, especially in the case of parallel implementations of these codes:

$$FOM = \frac{r \times \left[-\log_{10}\left(\frac{CBER}{RBER}\right)\right]}{I_d \times T_d \times H_d}$$

In this expression, r is the code rate, $CBER$ and $RBER$ are the corrected and raw bit error rates, respectively, T_d is the time delay (in gate delays or clock cycles) for decoding, H_d is the hardware complexity of the decoder in number of gates or equivalent, and I_d is the interconnection complexity of the decoder in terms of the total length of connections between bit cells in a parallel implementation. The higher this figure of merit, the "better" the code. Note that the quantities in the numerator delineate the efficiency (code overhead, correctability) of the code, while the denominator measures the decoding hardware efficiency.

4.3 Simulator development

A software package was developed for the simulation and evaluation of page-oriented optical memory systems. The package is flexible enough and can be adapted to many different technologies. Any mathematical or analytical model for a storage medium and/or optical system can be programmed and added to the existing routines as well as any error control code. It can also be programmed to use any data recovery scheme that the user wishes to try on their data.

So far, the simulator has been used mainly to simulate a volume holographic memory system, since the raw data that were available to us came from our own and Caltech's holographic memories. In the following subsections we outline the functionality and the capabilities of the simulator and provide some representative examples.

The simulator is based on an image processing program called Khoros developed at the University of New Mexico. Khoros is a collection of image processing routines linked together by a visual user interface called Cantata, which allows users to treat routines as blocks. Each block can receive a two-dimensional input, perform an operation to the data, and generate an output that can be directed to another block. Blocks can contain many other blocks at different levels of nesting and can be arranged in any form of a directed graph. Khoros users can create their own routines and incorporate them into the visual interface.

4.3.1 Simulator – General overview

The simulator is developed within the *Khoros* system and can be executed in the Cantata's visual programing environment. The program can be described as a directed graph in which the nodes are operators or routines (functions) and the links between the nodes represent paths through which data can be transferred from one node to another. The visual program and other objects associated with it, such as *variables*, make up a *workspace* which can be loaded into Cantata's interactive execution environment as the *main* workspace and can be executed as a whole or in steps.

The main components of the simulator are its nodes. These can be either simple, that is, a single operator or function, or collections of these into a *procedure*. The operators are control flow constructs in the language, such as loops or conditionals, and the functions or routines are complete programs (usually written in C) developed under the Khoros system or provided by it. These routines can be found under certain *Toolboxes* and most of them perform like a *filter*. They read input data (e.g. in this case an array of bits or floating point numbers) and they produce output data that are the result of applying the algorithm the routine implements to the input data. For example, an *fft* routine would read an array of floating point numbers from its input and produce an array of complex numbers in its output that represent the discrete Fourier transform of the input data.

Figure 1 shows the top level visual program that implements the simulator. It is represented by five procedures: `Start`, `Data Encoding`, `Error Generation`, `ED/C Mechanism`, and `Stop`. These are connected by data paths such that data are passed on from the procedure `Start` to the procedure `Data Encoding`, from that to `Error Generation`, and so on, until the data reach the procedure `Stop`. There is also a connection between the procedures `Start` and `Stop` so that the input data can be compared with the output data. Therefore, input data generated in the procedure `Start` are encoded with the error detection and correction algorithm used in the procedure `Data Encoding`. Then the encoded data are subjected to various error processes as specified by the procedure `Error Generation` and are passed on for decoding and error correction to the procedure `ED/C Mechanism`. The corrected data along with the input data are finally passed on to the `Stop` procedure where statistics are gathered and the errors that were not corrected are displayed.

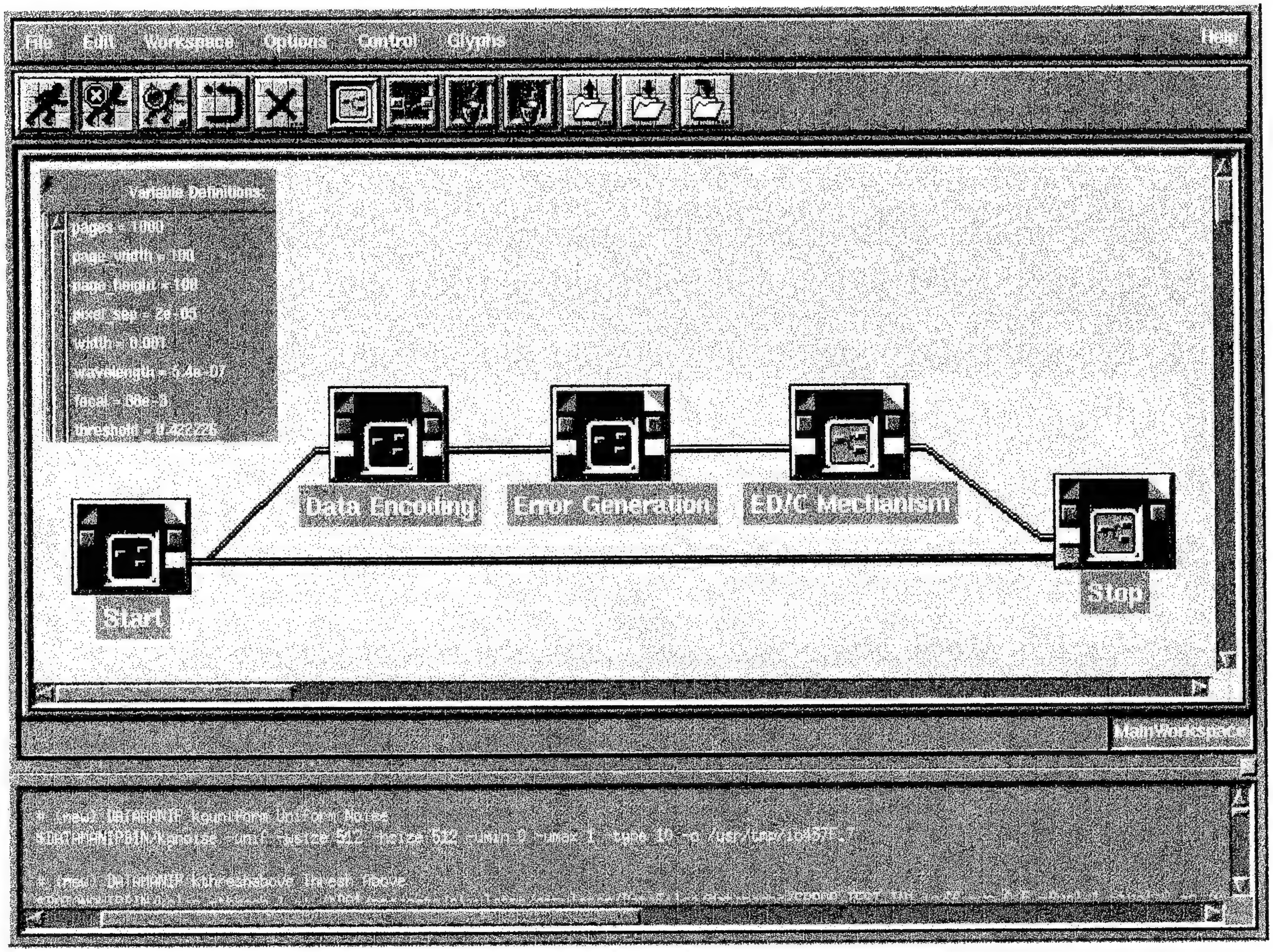
Before we proceed with describing each top level procedure separately, we should indicate that each procedure can be modified by the user so that a specific configuration that can be implemented by the provided software can be tested and results can be gathered. Therefore the procedures we will be describing serve primarily as examples or stubs. In addition, as more error generating processes, coding and decoding algorithms and other miscellaneous pieces of software are implemented, the simulator will be continue to evolve. Also, if the behavior of the simulated system needs to be studied over several pages of input data, the whole top level program can be embedded in a loop construct provided by the visual programming language.

The procedure `Start` generates an array of bit data which is passed on to the following procedure to serve as the array of information bits. The size of the array of data can be specified by appropriately setting the parameters of the routine that generates the data, which subsequently sets the program variables `Pagewidth` and `Pageheight`, which are used by subsequent routines. There are other ways of generating input data. For example, data that reside in a system file can be used instead, either directly or after being reformatted by appropriate khoros routines.

The array of bits generated by the procedure `Start` is then passed on for encoding to the procedure `Data Encoding`. In this procedure, the appropriate encoding algorithm for the system under study is used to produce an array of information and parity bits that will be stored in an optical memory and will be subjected to various error processes. The user can select from the Toolbox EC Coding/Decoding encoding routines. It should be noted that the size of the augmented array of bits is again passed on to program variables.

The output of the previous procedure, an array of information and parity bits, is then processed by the `Error Generation` procedure (Figure 2). The bits are converted to floating point values, since the optical memory system is primarily analog and various transforms are applied to the array of data in sequence to simulate various error processes. The simulator, at this stage, does not attempt to simulate the devices and the processes that implement the optical memory, but rather the effects of various error processes on the data values. For each error process we use an adequately precise theoretical model to calculate the output data values from the input. At the end of the pipeline, the floating point data values are converted back to bits. A thresholding function can be used to accomplish this or any other appropriate procedure that has been implemented. The bits will be passed on to the next procedure for decoding. The `Error Generation` procedure can be completely specified by the user, so that an appropriate model of an optical memory system can be simulated. The user can choose routines we have implemented from the Error Generation Toolbox or from standard Khoros toolboxes to specify a pipeline that would best match the optical memory system that is to be simulated. Thus our simulator is generic enough and flexible to allow the study of a variety of optical memory systems.

After the array of bits has been subjected to the error processes that simulate the optical memory system, it is processed by the decoding algorithm. This is done in the `ED/C Mechanism` procedure. The user can select from the Toolbox EC Coding/Decoding a decoding routine that corresponds to the encoding routine. The parity bits are checked and if errors that can be corrected are detected, they are corrected. Errors that are detected but cannot be corrected are reported. At this point statistics about the error rate of the system can be gathered. These and additional statistics can also be gathered at the following and last procedure `Stop`. It should be noted that the user is responsible for taking care that the coding and decoding algorithms correspond.



Variable Definitions:

```
pages = 1000
page_width = 100
page_height = 100
pixel_sep = 2e-05
width = 0.001
wavelength = 5.4e-07
focal = 88e-3
threshold = 0.422225
```

```
# cmaw2 DATAMANIP kguniform Uniform Noise
$DATAMANIPBIN/kgnoise -unif -wsize 512 -hsize 512 -umin 0 -umax 1 -type 10 -o /usr/tmp/io457E.7

# cmaw2 DATAMANIP kthreshabove Thresh Above
[illegible]
```

Figure 1. Top level view of the simulator.

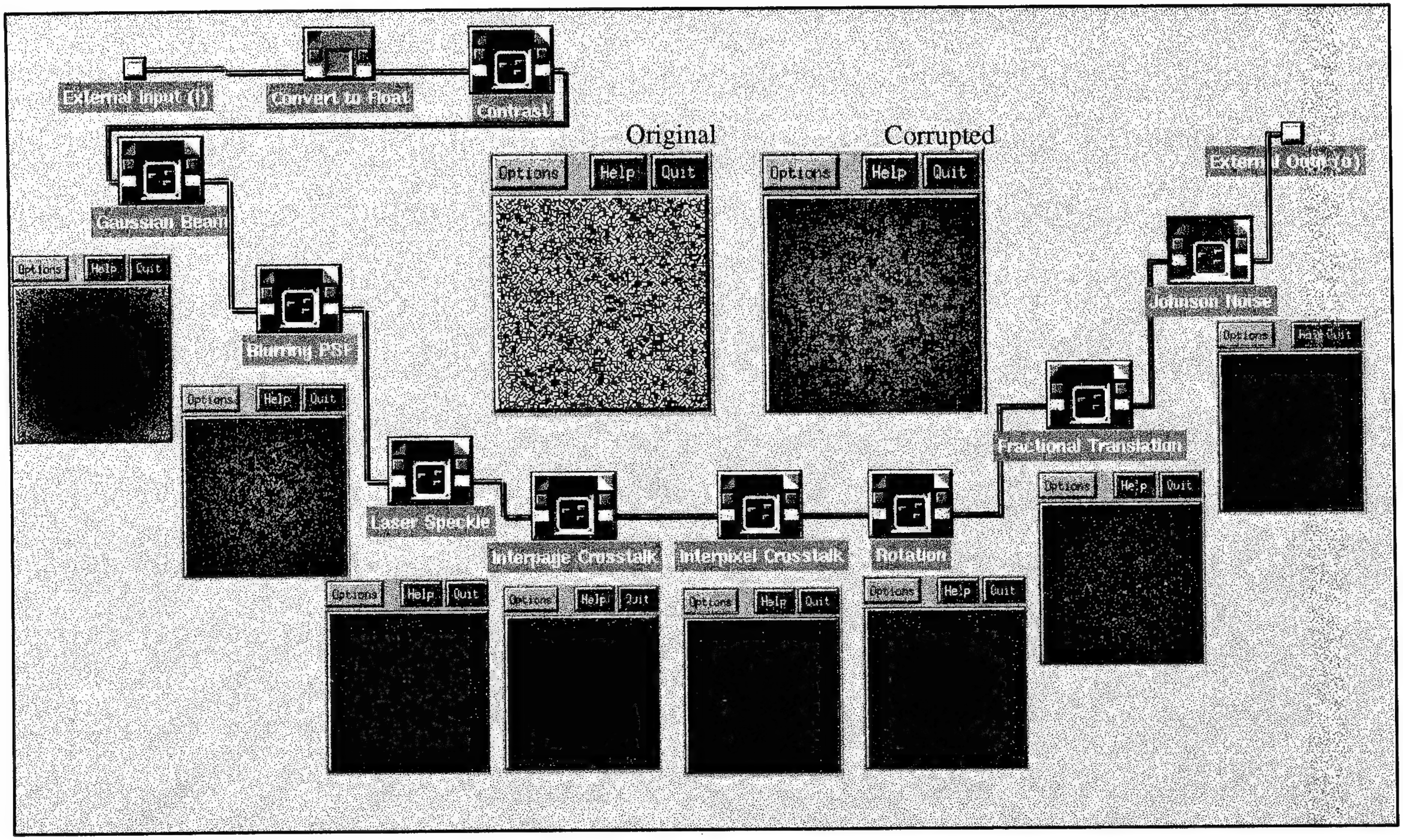
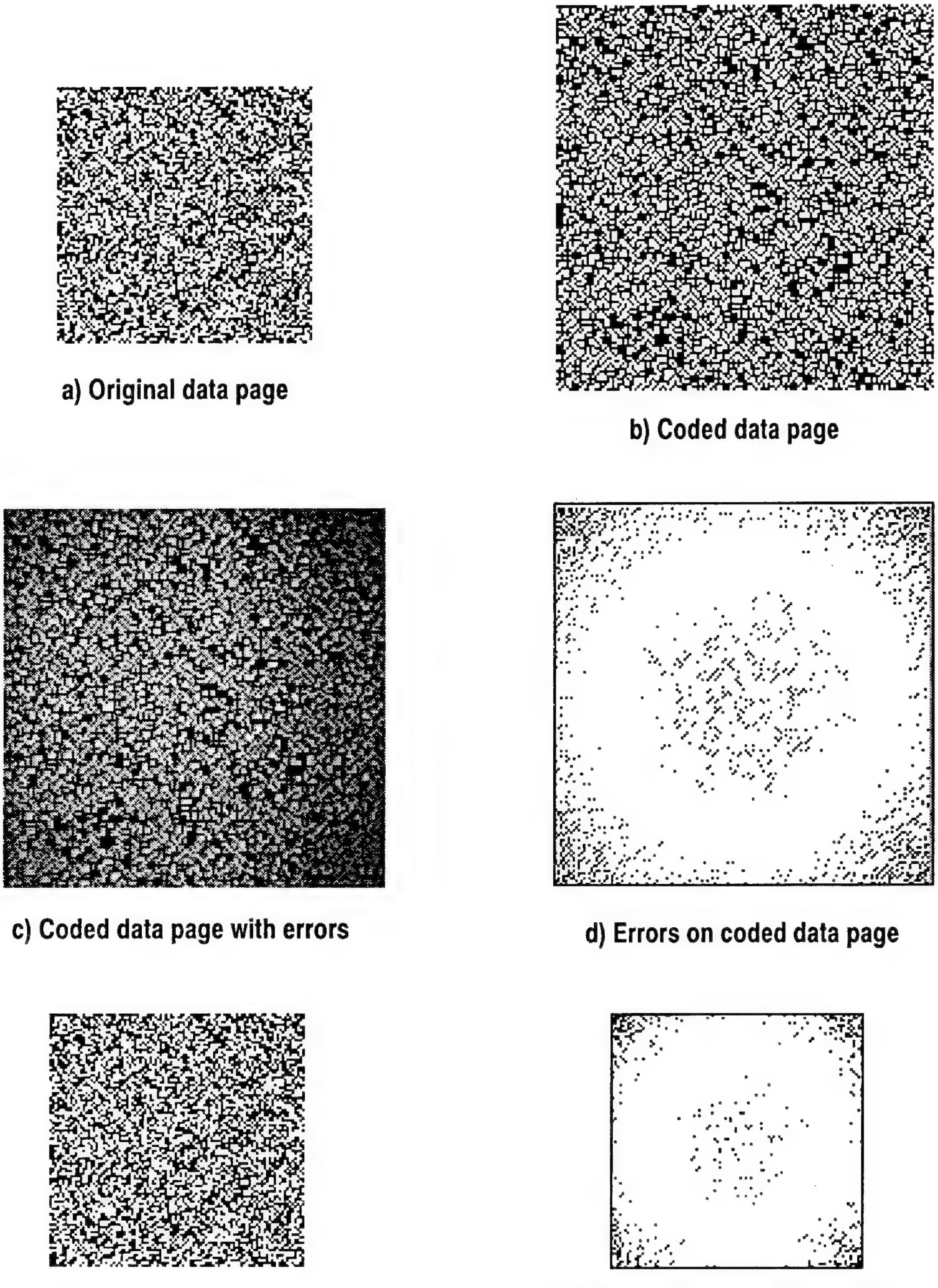


Figure 2. Error Generation procedure.



a) Original data page

b) Coded data page

c) Coded data page with errors

d) Errors on coded data page

e) Decoded data page

f) Errors on decoded data page

Figure 3. Example data pages at various moments of simulation.

Finally there is the procedure `Stop`. In this procedure statistics can be gathered, errors that were not corrected can be identified and the output of the optical memory and/or the errors in the data can be displayed. Thus in this and/or the previous procedure the RBER and the CBER can be computed from the input data to the system and the output data.

4.3.2 Example use of simulator

The example in Figure 3 illustrates the simulator's use in evaluating the performance of a code for correcting clustered errors. Parameters are entered that describe the memory system and nature of data. In this case, a data page of size 100 × 100 bits has been defined, shown in Figure 3a. A multiblock row and column coded data page is generated by the `Data Encoding` block and is shown in Figure 3b. The output of the `Error Generation` procedure is shown in Figure 3c, which depicts the corrupted coded data with the prominent errors sources being (in this example) the Gaussian beam profile and a translational misalignment of the detector. A hard decision at this time results in the distribution of errors shown in Figure 3d. Figure 3e depicts the results after performing the multiblock row and column decoding at the `ED/C Mechanism` block. The remaining errors in the corners and center of the page shown in Figure 3f represent clusters that have exceeded the correction capability of the code.

4.3.3 Error generation and error code blocks

Error generation is the heart of the simulation process. After the data have been encoded, the resulting image is subjected to corruption as close to the actual corruption that would occur in an optical memory system as possible.

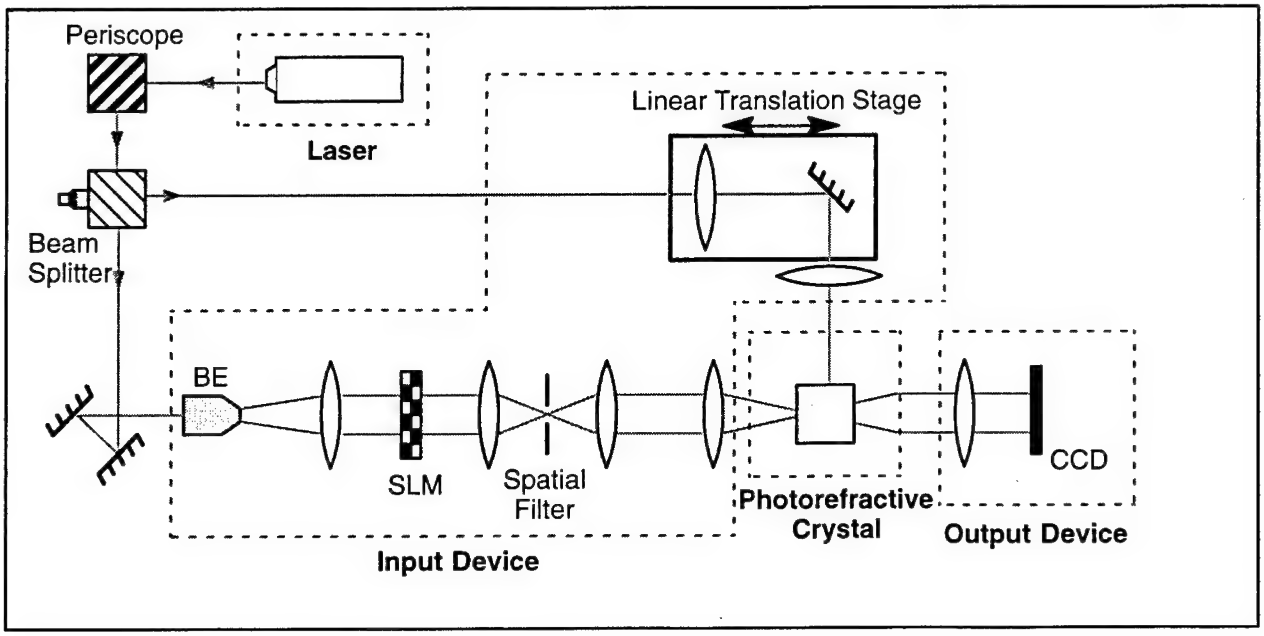


Figure 4. Volume holographic memory system.

The error generation block has grown during the course of this project, both in the internal components and the implementation considerations. We have created eight error blocks to describe common system errors. While these errors are currently tuned to the specifics of volume holographic memories, the simulator is not limited to a particular medium or technology. Any page-oriented memory system whose errors can be properly characterized can be coded into the simulator.

While the error generation block began as a simple error application block, our work has shown that this is the unit that must be optimized to achieve a reliable simulation of real data. Initially, errors were applied one at a time in a serial fashion. While this allowed us to do many things with theoretical error patterns, it did not produce good optimization results. In the future we will move to a combination of serial and parallel application of errors. Since this model is more complicated than the serial model, the optimization will be done using external routines.

Each of the error code blocks that have been constructed will be briefly described below. The current groupings of errors are presented in Figure 4 which depicts a volume holographic memory system: laser, input device, photorefractive crystal, output device. The four blocks are considered serial error processes while the errors inside each block are applied in parallel.

4.3.3.1 Laser source – Gaussian beam profile

The majority of errors from the laser source is dominated by the beam profile. We have coded a gaussian intensity profile whose parameters are the x and y variance and the x and y beam center. This covers the largest variety of possible sources.

4.3.3.2 Input device

Spatial light modulators (SLMs) continue to be the most popular input devices for optical memories. They allow large pages to be composed at one step but they tend to introduce several types of errors into the memory system. The most significant error, contrast ratio, is discussed in the next section. Another error source, multiple diffraction orders, can cause some power attenuation and the subsequent spatial filtering of high order terms gives rise to pixel blurring due to high frequency cut-off.

4.3.3.2.1 Contrast ratio

When the laser beam passes through the SLM, its amplitude and phase profile are modified. Amplitude modulation SLMs operate by turning some pixels 'on' and some pixels 'off'. Since neither the transmission nor the blockage of light is perfect, the output of the SLM is characterized by a contrast ratio. A contrast ratio of 30, for example, means that an 'on' pixel is 30 times brighter that an 'off' pixel. Very low contrast ratio means that all pixels are close to 50% of the incident light power.

4.3.3.3 Photorefractive crystal

Even though the properties of photorefractive crystals have been studied for many years, further investigation on the material properties is required. Errors which are relatively well understood at present include the diffraction efficiency of the stored holographic grating, speckle scattering at the crystal surface, interpixel crosstalk, edge effects or high frequency cut-off due to finite crystal size (blurring), and interpage crosstalk. This is arguably the most complex and most crucial segment of the memory system.

4.3.3.3.1 Diffraction efficiency/attenuation

Diffraction occurs when the recorded holographic grating reconstructs the initial data beam. As more pages are recorded in the medium, the percentage of diffracted light drops to relatively low levels. Most systems work in the 10^{-4} and lower range. This attenuation of the input reduces the noise margins due to the difficulty in detecting very low powers. We represent this effect as a constant attenuation coefficient.

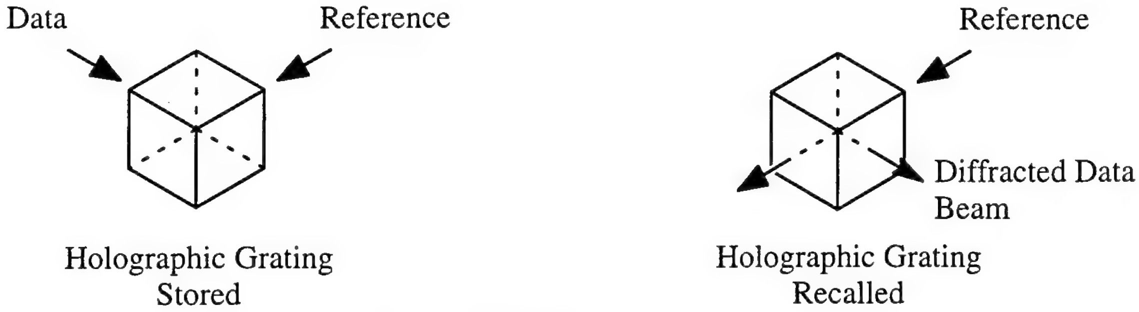


Figure 5. Diffraction effects.

4.3.3.3.2 Speckle scattering

As both the image and reference beams propagate through the optical system, they will pass through and be reflected by several surfaces. Rough surfaces with features in the order of a wavelength (e.g. crystal faces, SLM, etc.), will distort the wavefront and will cause further scattering with varying degrees of phase lag. Propagation of this light results in the addition of these various scattered components with varying delays. Interference of these dephased but coherent wavelets results in the granular pattern known as speckle. This is a multiplicative noise described by [2]:

$$I_{tot}(x, y) = a * I_{sig}(x, y) * I_s(x, y) \tag{1}$$

where,

I_{tot}: total intensity distribution
I_{sig}: signal distribution
I_s: effects from imperfect surface
a: proportionality constant

The simulator handles speckle noise by generating random dephased amplitude spread functions that are determined by the specification of a height function. A random process, which requires a mean and variance as input, is used to specify the height function, characterizing the rough surfaces.

4.3.3.3.3 Interpage cross-talk

In volume holographic memories a read-out image contains contributions from all pages that have been stored in addition to the page being referenced. When dealing with an angularly multiplexed scheme, the angular dependence of the diffracted wave amplitude follows a sinc function. The reference for that hologram will be Bragg matched to the sinc's main lobe, but at the same time it will impinge upon the side lobes of other holograms which are stored. These Bragg mismatched contributions are responsible for the interpage crosstalk.

The amount of interpage crosstalk is related to the number of holograms stored, size of the recording medium, dimension of output plane, focal length of the lenses, and wavelength of light. The noise to signal ratio (NSR) can be approximated by the following equation where the NSR is defined in a straightforward sense with the value being the direct ratio between noise and signal [4]:

$$\frac{noise}{signal} \approx \sum_{m \neq i}^{M} sinc^2\left[\frac{t}{\lambda}\,\frac{y_m - y_i}{f}\left(1 - \frac{y_2}{f}\,\frac{y_m + y_i}{2f}\right)\right] \tag{2}$$

$$\left\{ y_m = m\frac{\lambda f}{t \sin \theta} \right\} \tag{3}$$

where,

M: number of holograms, $(N-1)/2$

m: $\in \; [-M, -(M-1), .., 0, .., (M-1), M]$

t: thickness of medium along z direction

f: focal length of lenses in the system

$y_m,\ y_i$: position on reference plane; y_i : page referenced,
y_m : pages not referenced

λ: wavelength of light

d: linear dimension of output plane, equal to $2y_{2max}$

The simulator adds noise to a signal by calculating the NSR from equation (2) on a pixel per pixel basis and generates the error distribution. The input required for this error consists of a number indicating how many *neighborhood* pages to consider along with the various system parameters.

4.3.3.3.4 Interpixel Cross-talk

Interpixel cross-talk is due to the blurring caused by the finite size of the recording medium. Each square pixel of the data page becomes a convolution of a square and two sinc functions in the field that is received by the detector, shown in the following equation:

$$U_{p(0,0)} = A_0 \int sinc\left(\frac{D}{\lambda f}x\right) rect\left(\frac{x}{\delta}\right) dx \times \int sinc\left(\frac{D}{\lambda f}y\right) rect\left(\frac{y}{\delta}\right) dy \tag{4}$$

The energy of the signal that falls outside of the main lobe is noise, and by nature of the sinc functions its main contributions lie in the same row or column as the pixel being considered. The transverse size of the crystal (D), wavelength of light (λ), and focal length of lenses in the system (f) influence the degree of stretch in the sinc functions. A SNR can be computed for a single pixel by [6]:

$$SNR = \left(\frac{3}{4}\frac{D^2 S^2}{\lambda^2 f^2}\right)^2; \qquad S : pixel\ separation \tag{5}$$

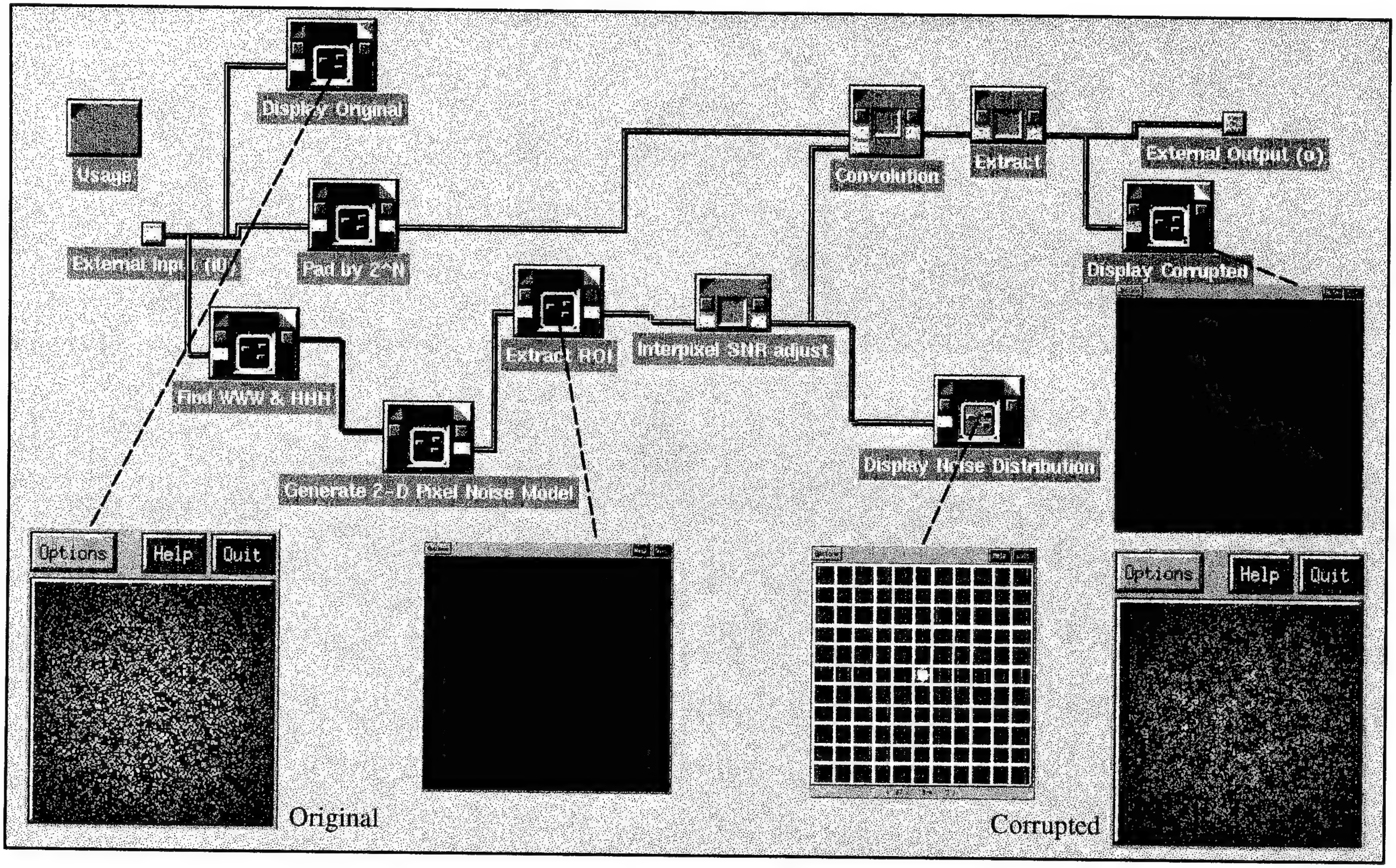


Figure 6. Interpixel Crosstalk procedure.

The simulator takes into account that the SNR can be isolated into a single calculation for an individual pixel. The interpixel crosstalk procedure is shown in Figure 6. The noise distribution around a pixel is generated by the approximation of Equation (4). From this, a region of influence is extracted by removing the areas where the contributions are negligible. The sinc-like plot is from a scan line sample showing how a pixel's energy is distributed amongst nearby pixels that lie along the same row. The noise distribution is also shown with a grid identifying the components that fall within each pixel location. The values inside each pixel region are averaged, and then the noise levels are adjusted to meet the SNR criterion of Equation (5). The resultant distribution is convolved with the data page in order to generate the intrapage interpixel cross-talk error. The original data page and resultant corrupted data page are also shown in Figure 6. A histogram accompanies the corrupted data page image to show the resultant spreading of the dark and light pixel distributions.

4.3.3.4 Output device

The output device is where the recreated page of data is detected and converted to electronic form. Current systems utilize charge coupled devices (CCDs) to sample the light level and output the page with no additional processing. Any detection or correction would have to be implemented in hardware during the transfer to an electronic host computer or in software on the computer itself.

Smart photodetector arrays (SPAs) are a concept we have proposed as a way to detect the light and process the data before they are transferred to the host computer. By integrating a photodetector and some logic, we can perform simple error correction schemes in parallel before the data are passed to the electronic computer. This also helps increase the bandwidth of our system because parity bits do not need to be transferred.

There are several types of errors that occur in optical systems in general. Of these misalignment, rotation, defocusing errors, and electronic noise are most significant at the detector array even though these problems could be a factor in other parts of the system. In general, we assume a relatively well aligned system. This will be true in all practical optical memory systems.

4.3.3.4.1 Translational misalignment

Misalignment is a system consideration for all optical systems and can cause incorrect imaging onto the detector array. We have studied partial pixel misalignment due to translations in both the x and y directions. As expected, large misalignment (50% of the pixel width or more) will we unacceptable. Smaller degrees of misalignment when coupled with other errors can produce some problems.

4.3.3.4.2 Rotational misalignment

Rotation is another kind of misalignment problem. Where a translation affects all pixels equally, rotation affects pixels further from the rotational axis more severely. Small angles can produce significant problems as the size of the page increases.

4.3.3.4.3 Defocusing (blurring)

Blurring comes from two sources: poor focusing and high frequency cut-off. When optical components, especially in a 4-f system required to do Fourier transforms, are not accurately positioned along the optical axis, small image distortions can be observed. This manifests as a decrease in the resolution of the optical system. In addition to the focal plane alignment, each optical component, due to its finite size, can block high frequency components of the optical signal. This is similar to low pass filtering and may lead to a reduction of contrast. Both effects result to a blurred image.

4.3.3.4.4 Electronic noises

The dominant noises that play a role in CCDs and other photodetector arrays—transfer noise, sensing noise, pattern noise, shot noise, Johnson noise, thermal noise, etc.—are additive noises that corrupt the data page in the last stage of the error generation process. From the device's specifications and its operating conditions the appropriate statistical noises can be computed and then added to the image.

4.4 Uses for the simulator

The Khoros simulator has proven a valuable tool for looking at real data and estimating error effects. It allows pages to be viewed as they would be seen through the system. Error distributions of different types can be easily viewed, stored, printed, and manipulated. This section will briefly review some of the areas we have studied during the course of this project and some of the useful tools we have developed.

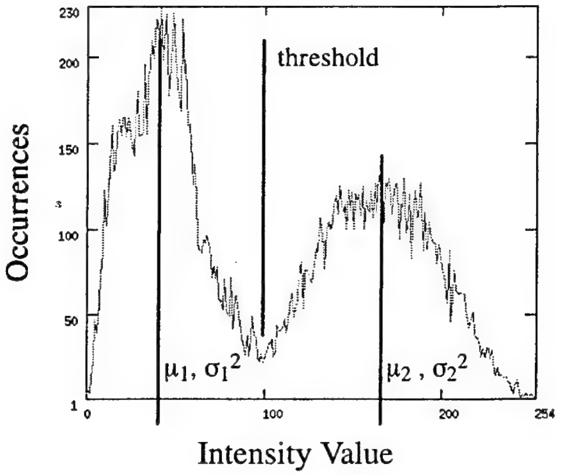


Figure 7. Intensity histogram.

4.4.1 Intensity histograms

Procedures have been created that automate the process of generating histograms of the intensity values for a given data page. Information about the underlying random processes can be obtained from the histogram by examining the probability distributions for the light and dark pixels. This is an approach of uncovering which type of noise is most dominant—whether it follows a Gaussian, exponential, Rician, or some other probability distribution. The separation point between the probability distributions represents the threshold for distinguishing the logic-high and logic-low values. When statistics are taken for the separated intensities, a SNR can be calculated from:

$$SNR = \frac{\mu_2 - \mu_1}{\sqrt{\sigma_1^2 + \sigma_2^2}} \tag{6}$$

where μ_1, σ_1^2 and μ_2 , σ_2^2 are the mean and variance for the dark and light PDFs, respectively.

4.4.2 Data extraction

For most of the representations of a data page, the simulator treats the continuous image as a pixelated image with each pixel representing the average intensity over any particular data bit. In this structure, a 1000 × 1000 data page has a 1000 × 1000 intensity level representation. This facilitates the complex operations that are performed. One-to-one matched system will provide this sort of output, but most systems employ SLM and CCD over-sampling where several pixels are used to represent a single data bit, as shown in Figure 8.

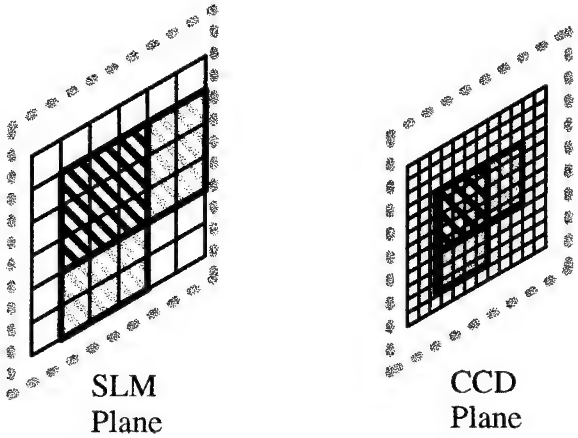


Figure 8. Example of 3×3 SLM "mega-pixel" over-sampled by CCD.

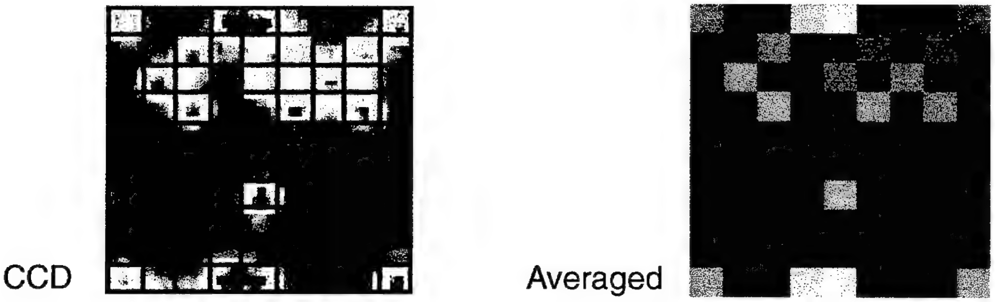


Figure 9. Data extraction and averaging.

Various data extraction aids extend the capabilities of the simulator which are able to angularly align a page, if necessary, and then extract regions of data bits. An averaging operation is performed upon these regions to achieve a single intensity per bit. An example of data averaging for data extraction is shown in Figure 9.

4.4.3 Testing error codes

Using the encoding/decoding blocks, we studied the power of different error correcting codes. For example, we studied the multi-block row and column code and the self-orthogonal code based on the Smith construction. Using the same error parameters, we created pages which had errors of almost the same distribution and quantity. We then decoded the corrupted page, compared it to the original page, and compared the results of the two codes. After repeating this process with several slightly modified error parameters, we were able to confirm some of the results we had predicted with our “figure of merit”.

4.4.4 Real system approximation and simulator optimization

Much of our simulator effort has focused on accurately describing real systems. Given the input and output of a specific optical memory system, the simulator must be tuned to match output when presented with the input. This work gives us two very valuable tools. First, we can look at the parameters of the simulator after we have matched the outputs and receive some data regarding which error processes are most prevalent in that particular memory. Second, we can run experiments in the simulator to predict what the output of different configurations of the memory might be if error parameters are modified.

One example of modeled system was our own experimental holographic memory setup. We currently record and retrieve holograms using a lithium niobate crystal. When presented with the input pattern, the simulator was optimized to produce an output fairly close to the real data.

The method of tuning the error parameters turns out to be an optimization problem. Each error process contributes to the overall output. We believe that the four main categories of errors should be four serial error blocks. The errors in the input device affect all of the errors in the crystal. However, we believe that the errors inside these block should be applied in parallel. For example, the interpage crosstalk in the crystal is independent of the diffraction efficiency. While there may exist some correlation between these errors, we believe an assumption of independence will not be too constraining for acceptable optimization results. Work in this area continues.

4.4.5 Footprints

The simulator has also helped us in our search for a way to distinguish between different errors. The problem arises when real output data pages are studied and we need to identify the most significant error in the system. We have used the simulator to look at the transforms of single error patterns. For example, errors due to contrast ratio produce a different spatial pattern when Fourier transformed than blurring noise. Not all patterns, however, are separable using the Fourier transform. Blurring, crosstalk, rotation, and translations have very similar patterns in Fourier space. Other transforms may allow a further distinction of these errors.

Accurate footprints would provide another tool to analyze the data from real systems. They would allow the most significant error in an optical system to be identified and addressed.

4.4.6 References

[1] J.C. Dainty, *Laser Speckle and Related Phenomena*, Springer-Verlag, New York, 1975.

[2] C. Gu, J. Hong, I. McMichael, and R. Saxena, “Cross-talk-limited storage capacity of volume holographic memory,” *J. Opt. Soc. Am.*, Vol. 9, No. 11, Nov. 1992.

[3] X. Yi, P. Yeh, and C. Gu, “Statistical analysis of cross-talk noise and storage capacity in volume holographic memory,” *Optics Letters*, Vol. 19, No. 19, Oct. 1, 1994.

4.5 Evaluation of existing array coding schemes for use in POMs

Our search of the literature yielded a large variety of error control schemes that may be suitable for POM systems. A representative list is given in Table 2. Note that this is only one of several possible classification schemes.

Table 2. Classification of Error Control Codes for Memory Systems

1. **Codes Designed for Random Errors**
    - SPC (Single Parity Checksum)
    - CRC (Cyclic Redundancy Check)
    - Repetition
    - RAC (Row and Column)
    - Wing code
    - Hamming
    - Reed-Solomon (RS)
    - Reed-Muller (RM)
    - Orthogonal, Bi-Orthogonal, Simplex
    - Self-Orthogonal
    - Other similar codes
        - Complex-rotary techniques
        - DBBD (Differential balanced block designs)
        - SBIBD (Symmetric BIBD)
    - BCH (Bose-Chaudhuri-Hoequenghen)
    - Golay
2. **Burst Correction Codes**
    - Single-burst error-correcting (SBC)
    - Multiple-burst error-correcting
3. **Burst Correction through Convolutional or Orchard Codes**
    - RBC (Row Burst Correction / V-checking)
    - SEC Block Convolutional Codes
    - Difference Triangles
    - Scott and Goetschel Difference Triangle
    - Cross Interleaved Code
    - CPC (Cross Product Code)
    - Wyner-Ash
4. **Cluster Correction**
    - Interleaved RAC
    - Concatenated
        - Block-block (RS)
        - Convolutional-block (RS)
        - Cross-interleaved (for bursty channels)

- Coded modulation
  - BCM (Blocked Coded Modulation)
  - TCM (Trellis Coded Modulation)

4.5.1 Evaluated array codes

4.5.1.1 Row-and-column (RAC) array code

This is the simplest array code [1]. Figure 8 shows that this code consists of an array of $k_1 \times k_2$ information bits along with row and column parity check bits ($n_1 = k_1 + 1, n_2 = k_2 + 1$). This code is capable of detecting any two errors and correcting any single error. If there are more than two errors present, the code may not detect all possible cases and it can also miscorrect and misdetect in certain cases.

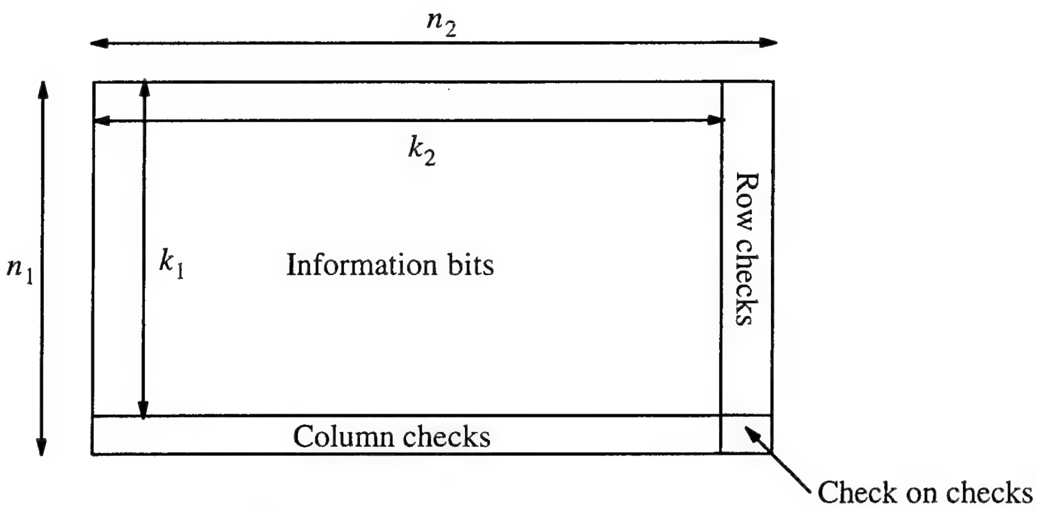


Figure 8. Row-and-column array code.

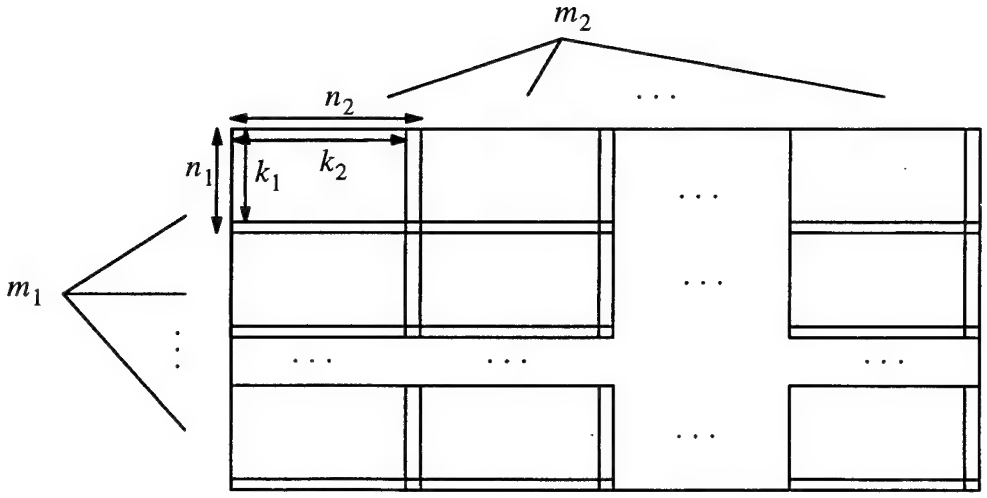


Figure 9. Multiblock row-and-column array code.

4.5.1.2 Multiblock row-and-column array code

For the RAC code to be of any significant practical application, the array of bits must be small. As a solution to this problem for larger sizes, the array of information bits can be subdivided uniformly into smaller arrays and each can be encoded using the RAC code. Therefore, we have an $m_1 \times m_2$ array of $n_1 \times n_2$ RAC "codewords" or codeblocks, as shown in Figure 9.

4.5.1.3 Self-orthogonal block array code

The self-orthogonal block array code [2] is a representative of a class of codes that are capable of correcting a larger number of random errors than the simple row-and-column (RAC) code. In fact, the RAC code is a special case of an orthogonal code where there is a set of two orthogonal parity check equations on each information bit of the array. In general, if there are J orthogonal parity checks on any information bit in the array, correction of $\lfloor J/2 \rfloor$ errors is possible. Correct decoding can be performed, if at most $\lfloor J/2 \rfloor$ errors occur in the array, where an information bit is corrected if the majority of the J parity check bits orthogonal on it fail.

The self-orthogonal block array code can be obtained by means of the *Smith construction* as follows. Let p be prime (or a power of a prime) and let the rows and the columns of the $p \times p$ information bit array be labeled $0, 1, \ldots, r, \ldots, p{-}1$ and $0, 1, \ldots, c, \ldots, p{-}1$, respectively. Then, with each information bit at position (r, c) we associate $\delta{-}1$ numbers, $D_i(r, c) = (r + ic) \bmod p$, where $0 \le i \le \delta{-}2$ and $\delta \le p + 1$. It has been shown that no two information bits have more than one common value of $D_i(r, c)$ and therefore each value of $D_i(r, c)$ defines a set of orthogonal parity check equations.

4.5.1.4 A family of cluster-correcting array codes

This is a family of codes that uses staggering across rows and columns in order to achieve a single cluster correction [3]. Any number of errors that occur within a rectangular block of bits can be corrected with such a code. However, only a single cluster can be corrected with this code. The size of the rectangular block that would contain the errors we wish to correct specifies limits on the size of the block of this code. Let the size of the rectangular cluster (block) be $b_1 \times b_2$. Then the size of the array code block is $n_1 \times n_2$, where $n_1 \ge 2b_1b_2 - b_1$ and $n_2 \ge 2b_1b_2$ and b_1 divides n_1 and b_2 divides n_2. The size of the block of information bits is $k_1 \times k_2 = (n_1 - 1) \times (n_2 - 1)$ and a RAC-like array of information and row and column parity bits is produced before staggering. The staggering of the array is performed by first rotating the bits of rows and then the bits of columns. For each row i, a shift to the right by $b_2(i \bmod b_1)$ places is applied and then for each column j, a shift down by $b_1(j \bmod b_2)$ places. In this way the code bits are sufficiently interleaved so that a cluster of errors will affect several row and column parity bits, with no error within the block affecting more than a single row and a single column parity bit. In this way, the location of the cluster and the errors in it can be determined so that correction can be performed.

4.5.1.5 A single burst-correcting array code

This is also a code based on the RAC code. The burst-correcting capability of this code is achieved by a special diagonal *read-out* of the array (Figure 10), so that each error bit will affect a unique set of a row and a column parity check (and no other error bit will affect those parity checks) but the location of the errors will be correctly identified by the read-out sequence, assuming that bursts occur across rows [4]. Note that without the read-out sequence the RAC code cannot associate a row parity check failure with a corresponding column parity check failure, if more than one occur.

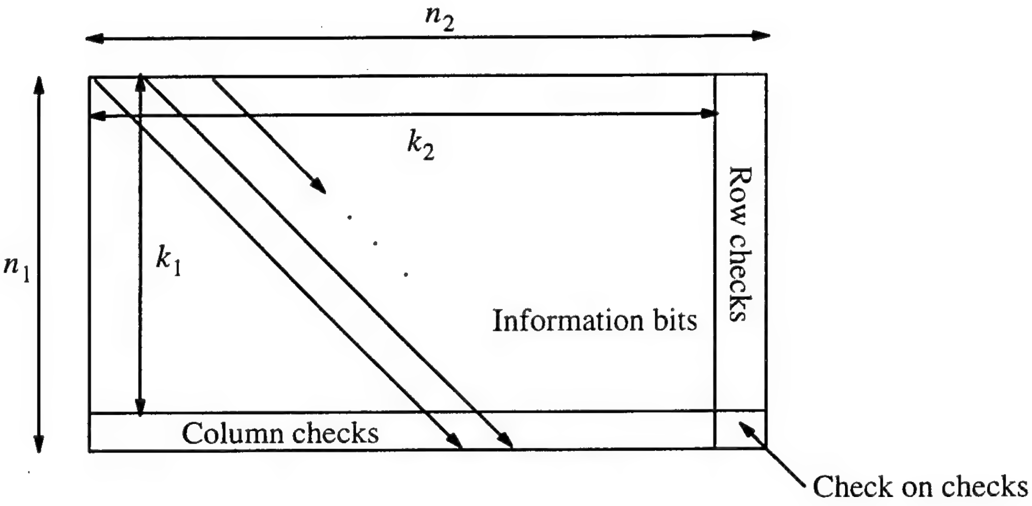


Figure 10. Single burst-correcting code and read-out pattern.

For such a code to be able to correct a burst of specified length, certain conditions on the dimensions of the array must be met. In order to correct a burst of k_1 bits, the size of the array of information bits must be $k_1 \times k_2$, where $k_2 \geq 2(k_1 - 1)$. The read-out starts from the top left bit in the array and progresses along the main diagonal. After reaching the bottom row of the array the read-out continues with the following diagonal, i.e. starting with the bit next to the top left. The read out continues in this way, assuming a cyclical array (i.e. the last column is followed by the first), until all the bits have been processed. In other words, after the check on checks bit has been processed, the diagonals wrap around the array.

4.5.1.6 Wing and multiblock wing code

The wing code is a triangular code similar to the RAC code [5]. It consists of a set of information bits that form a right angle triangle with a row of parity checks forming the base of the triangle (Figure 11). It is more efficient than the RAC code, i.e. it uses less parity bits, but it does not have double error detection capability. The wing code is a single correcting code that uses two orthogonal parity checks on every information bit. The difference is that each parity bit checks both a "row" and a "column" and in this way the wing code achieves a better code rate than the RAC code. Because it has a triangular form, it does not match the rectangular form of a page of optical memory. However, it might be useful as part of a coding scheme that uses it to protect the corners of the page and leaves the center to another coding scheme. In any case, the wing code might also need to cover a rectangular array of bits. To accomplish this, either the information and parity bits need to be rearranged into a rectangle or a multiblock scheme must be used. If the code block is rearranged into a rectangle, the simple straight interconnections of the triangular block are sacrificed and a more complex interconnection scheme is required. However, if two triangular blocks are used to form a rectangular block (Figure 12) the simple interconnection pattern of this code is retained and the rectangular form requirement is satisfied. The only problem is that the resulting rectangular block must be of size $n \times (n + 1)$. Therefore it is not possible to form an arbitrary size rectangular block by combining wing code blocks. However, similar restrictions on the size of the code block are required by several other codes, such as the cluster and burst correcting codes we have previously evaluated. It should be noted that this multiblock arrangement is different than the multiple-wing code which attaches several information wings to the same row of parity checks (along with a separate single parity check bit for each wing of information bits).

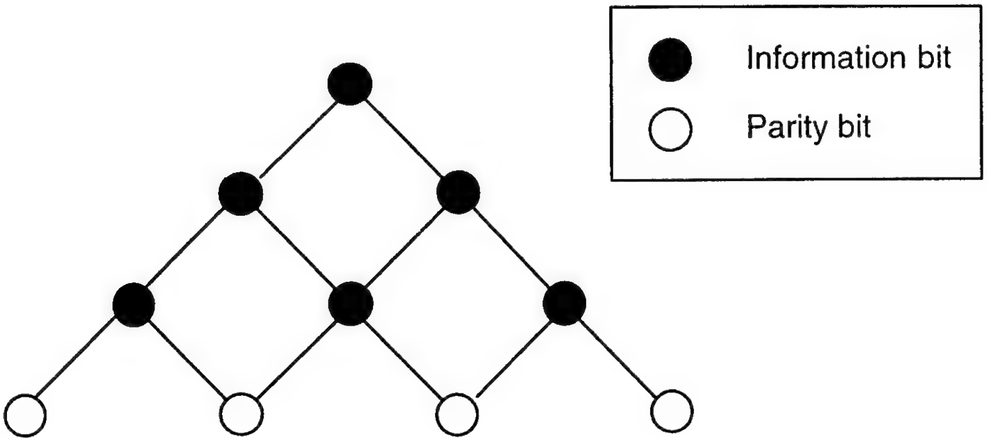


Figure 11. A (10,6,3) wing code.

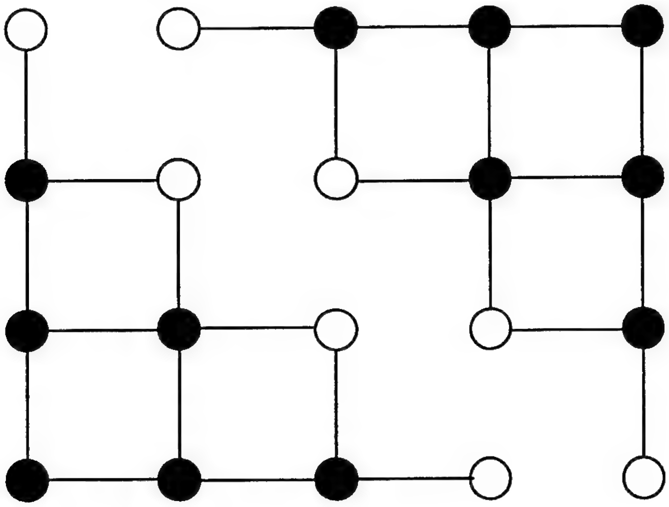
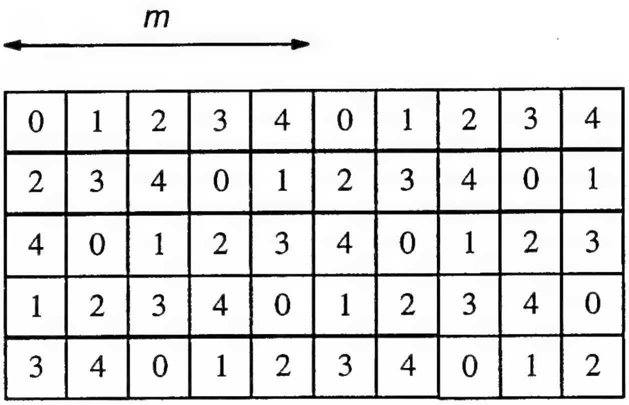


Figure 12. A 4 × 5 rectangular block formed by two wing code blocks.

4.5.1.7 Interleaving coding scheme for cluster error correction

Interleaving schemes for cluster error correction are attractive because they are based on a simple concept and are flexible. The choice of the degree (or depth) of bit interleaving controls the size of the cluster of errors that can be corrected and the choice of the component code controls the number of separate clusters that can be corrected (e.g. using a double error-correcting code, two clusters of errors can be corrected). This particular scheme that we evaluate was developed by Blaum and Bruck [6], and has the property that, considering the array of bits as an image and borrowing terms from image processing, the cluster is not restricted to a rectangular shape but it can be any 4-connected region in the array composed of a number of pixels/bits less than or equal to the number of error bits the scheme can correct.

The interleaving scheme is simpler than that of the previous cluster correcting code we have evaluated, however that code interleaved the bits of a single RAC code block by staggering the rows and the columns of the array, while this scheme interleaves the bits of several code blocks, by staggering only the rows of the array. Figure 13 shows an example of this interleaving scheme. Array positions with the same number represent bits that belong to the same code block. Depending on the size of the array and the number of errors in a cluster we want to correct, a code block may span a single column or it may span several columns (two in the example of Figure 13). In the case the code blocks span several columns, before staggering, the first one spans the 1st, n+1, 2n+1, ... columns, the second code block, the 2nd, n+2, 2n+2, ..., and so on. The number of code blocks, m, needed to correct any cluster of e errors must be $\frac{e^2}{2}$ if e is even, or $\frac{e^2+1}{2}$ if e is odd. This number m is called the *degree of interleaving* and the array is called e-interleaved. Similarly, if e is even, row i must be staggered by $(i \bmod m) \times (e - 1)$ places, while if e is odd, by $(i \bmod m) \times e$ places, where $i = 0, 1, \ldots$ and $(e-1)$ and e respectively must be relatively prime with m in each case. It should be noted that the number of rows and the number of columns of the array can be arbitrary, that is, they do not need to be multiples of m, as in the example of Figure 13, but they need to accommodate the size of the component code block and the interleaving requirements.



0	1	2	3	4	0	1	2	3	4
2	3	4	0	1	2	3	4	0	1
4	0	1	2	3	4	0	1	2	3
1	2	3	4	0	1	2	3	4	0
3	4	0	1	2	3	4	0	1	2

Figure 13. A 3-interleaved array with degree of interleaving 5.

4.5.1.8 References

[1] P. G. Farrell, "A Survey of Error Control Codes," *European Journal on Telecommunications*, Vol. 3, No. 5, pp. 441-453, 1992.

[2] R. J. G. Smith, "Easily-Decoded Efficient Self-Orthogonal Block Codes," *Electronics Letters*, Vol. 13, No. 7, pp. 173–4, 31 March 1977.

[3] M. Blaum and P. G. Farrell, “Array codes for cluster-error correction,” *Electronics Letters*, Vol. 30, No. 21, pp. 1752–3, October 1994.

[4] P. G. Farrell and S. J. Hopkins, “Burst-error-correcting array codes,” *The Radio and Electronic Engineer*, Vol. 52, No. 4, pp. 188–192, April 1982.

[5] S. Harari, “Fuzzy correction capability,” *Algebraic coding theory and applications.* (ed. G. Longo), Springer-Verlag, 1979, pp. 497-512.

[6] M. Blaum and J. Bruck, “Interleaving schemes for two-dimensional cluster errors,” Proceedings of DEPOM ’96.

4.5.2 Results of the evaluation

The parameters for the evaluation of the array codes can be roughly divided into two sets. A set of coding related parameters and a set of hardware implementation related parameters. Because the corrected bit error rate requires special attention, it is presented separately from the other coding related parameters.

4.5.2.1 Coding parameters

We summarize the results we obtained from our evaluation of a set of eight array codes in the following tables.

Table 3. Coding parameters (part a).

Code	*Length* *n*	*Information bits, k*	*Rate* *r*	*Coding overhead*
RAC	$n_1 \times n_2$	$(n_1 - 1) \times (n_2 - 1)$	$1 - \frac{n_1 + n_2 - 1}{n_1 n_2}$	$n_1 + n_2 - 1$
MBRAC	$m \times n'$	$m \times k'$	r'	$m \times (n' - k')$
Self-orthogonal	$p^2 + (\delta - 1)p$	p^2	$\frac{p^2}{p^2 + (\delta - 1)p}$	$(\delta - 1)p$
Cluster	$n_1 \times n_2 \geq (2b_1 b_2 - b_1) \times (2b_1 b_2)$	$(n_1 - 1) \times (n_2 - 1)$	$1 - \frac{n_1 + n_2 - 1}{n_1 n_2}$	$n_1 + n_2 - 1$
Single burst	$n_1 \times n_2 \geq n_1 \times (2n_1 - 3)$	$(n_1 - 1) \times (n_2 - 1)$	$1 - \frac{n_1 + n_2 - 1}{n_1 n_2}$	$n_1 + n_2 - 1$
Wing	$\frac{p(p + 1)}{2}$	$\frac{p(p - 1)}{2}$	$\frac{p - 1}{p + 1}$	p
MBWing	$m \times p' \times (p' + 1)$	$m \times p' \times (p' - 1)$	r'	$2 \times m \times p'$
Interleaving scheme	$m \times n'$	$m \times k'$	r'	$m \times (n' - k')$

Table 4. Coding parameters (part b).

Code	*Max. correctable errors*	*Max. detectable errors*	*Types of errors*
RAC	1	2	random
MBRAC	$\{\min = 1,\ \max = m\}$	$\{\min = 2,\ \max = 2m\}$	random
Self-orthogonal	$\left\lfloor \frac{\delta - 1}{2} \right\rfloor$	$\left\lfloor \frac{\delta - 1}{2} \right\rfloor$	random
Cluster	$b_1 b_2$	$b_1 b_2$	cluster
Single burst	$n_1 - 1$	$n_1 - 1$	burst
Wing	1	1	random
MBWing	$\{\min = 1,\ \max = 2m\}$	$\{\min = 1,\ \max = 2m\}$	random
Interleaving scheme	e	$2e$	cluster

In the above tables, n_1 and n_2 are the rows and the columns of the codeblock, respectively. In the case of multiblock codes, m is the number of code blocks that constitute the array, except in the case of the multiblock wing code where it is the number of pairs of code blocks (i.e. half the number of blocks). In the case of the interleaving scheme for cluster error correction, m is $e^2/2$ if e is even, or $\left(e^2 + 1\right)/2$ if e is odd. Also in the case of multiblock codes, primed parameters represent the parameters of the component single block code. In the case of the self-orthogonal code, p is a prime number (it can also be a power of a prime number) and $\delta - 1$ is the number of orthogonal parity checks per information bit. On the other hand, in the case of the wing code, p is the number of parity bits. In the case of the cluster-correcting code, the size of the rectangular block of errors that can be corrected is $b_1 \times b_2$.

4.5.2.2 Hardware parameters

All the codes we have evaluated can be decoded in serial or in parallel and no code requires read-out of multiple pages. The remaining results of the evaluation of the hardware implementations of the selected codes are summarized in the following tables. The above remarks for several variables also apply to the following tables. In what follows, hardware complexity is measured in terms of logic gates and timing results in terms of gate delays, unless it is mentioned otherwise. Finally, the interconnection complexity of an error correcting code (for parallel implementations only) is defined as the average length of the connections emanating from a cell ($\overline{l_c}$) multiplied by the number of cells. In addition, in the case of multiblock codes, this number is augmented by the average length of the connections emanating from a block ($\overline{l_b}$), multiplied by the number of blocks. Thus,

$$I = n\overline{l_c} + m\overline{l_b}$$

Table 5. Algorithm complexity

Code	*Serial implementation*	*Parallel implementation*
RAC	$O(n)$	$O\left(\sqrt{n}\log(\sqrt{n})\right)$
MBRAC	$O(m \times RAC)$	$O(RAC + \log m)$
Self-orthogonal	$O(n)$	$O(\lceil \log_2 p \rceil)$
Cluster	$O(n)$	$O\left(\log(\max(n_1, n_2))\right)$
Single burst	$O(n)$	$O\left(\log(\max(n_1, n_2))\right)$
Wing	$O(k)$	$O(\lceil \log_2 p \rceil)$
MBWing	$O(2m \times Wing)$	$O(Wing + \log m)$
Interleaving scheme	$O(m \times RAC)$	$O(RAC + \log m)$

Table 6. Hardware complexity (serial implementation)

Code	*Hardware complexity*
RAC	(coding) 1 XOR gate + max(n_1, n_2) 1 bit memory + (decoding) coding hardware + 1 XOR + 1 AND + 1 OR gate
MBRAC	(coding) H_c(RAC) (decoding) H_d(RAC)
Self-orthogonal	(coding) (δ–1) p-stage shift registers + $O((\delta-1)p)$ XOR gates (decoding) (δ–1) p-stage shift registers + p^2-stage shift register + $O((\delta-1)(p + 1))$ XOR gates + (δ–1)-majority gate
Cluster	(coding) 1 XOR gate + max(n_1, n_2) 1 bit memory + + 1 $\max(n_1, n_2)$ cyclical shift register (decoding) 1 CPU + $O(n)$ memory
Single burst	(coding) 1 XOR gate + n 1 bit memory (decoding) 2 n-stage shift registers + $(k_1 + k_2 + 5)$ XOR gates + 3 misc. gates
Wing	(coding) 1 XOR gate + n 1 bit memory (decoding) 1 XOR gate + ($n + p$) 1 bit memory + + 1 XOR + 1 AND + 1 OR gate (condition evaluation)
MBWing	(coding) H_c(Wing) (decoding) H_d(Wing) + 1 AND gate + 1 bit of memory
Interleaving scheme	(coding) H_c (decoding) H_d

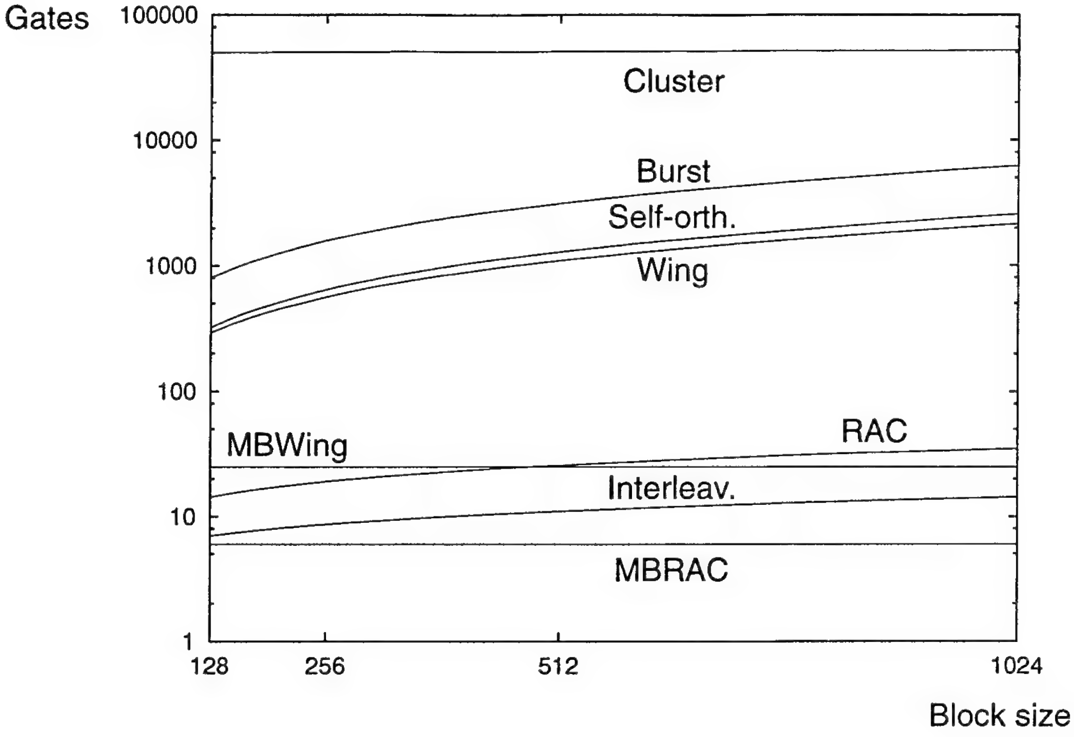


Figure 14. Hardware complexity (decoding, serial implementation).

Table 7. Hardware complexity (parallel implementation)

Code	*Hardware complexity*
RAC	(coding) $\left[[k_1(k_2-1) + k_2(k_1-1)] + \min(k_1 - 1, k_2 - 1)\right]$ XOR gates (decoding) $[[k_1(k_2-1) + k_2(k_1-1)] + \min(k_1 - 1, k_2 - 1) +$ $+ (k_1 + k_2 + 1)]$ XOR gates $+ k_1k_2$ (1 XOR + 1 AND gate) $+ (k_1 - 1) + (k_2 - 1)$ (1 XOR + 1 AND + 1 OR gate)
MBRAC	(coding) mH_c(RAC) (decoding) mH_d(RAC) + global AND hardware
Self-orthogonal	(coding) $(\delta-1)p\lceil\log_2 p\rceil$ XOR gates (decoding) $\left((\delta-1)p(\lceil\log_2 p\rceil + 1) + p^2\right)$ XOR gates $+ p^2$ $(\delta-1)$-majority gates + hardware for evaluating if correction is possible
Cluster	(coding) $\left[[k_1(k_2-1) + k_2(k_1-1)] + \min(k_1 - 1, k_2 - 1)\right]$ XOR gates + interconnections for row and column bit rotation (decoding) Coding hardware (for parity bit computation and bit rotation) $+ (k_1 + k_2)$ XOR gates + 1 CPU + $O(n)$ memory (for error detection and correction)
Single burst	(coding) $\left[[k_1(k_2-1) + k_2(k_1-1)] + \min(k_1 - 1, k_2 - 1)\right]$ XOR gates + interconnections for read-out rearrangement (decoding) Coding hardware (for parity bit computation and read-out rearrangement (reverse network)) + $+ (k_1 + k_2)$ XOR gates (for syndrome computation) $+ (\lceil\log_2 k_1\rceil + \lceil\log_2 k_2\rceil)$ OR gates (for error checking) + 1 $\lceil\log_2(\min(k_1, k_2))\rceil$-bit adder + $2n$ AND gates (for error correction)
Wing	(coding) $p(p - 2)$ XOR gates (decoding) $p(p - 2)$ XOR gate + $+ (p - 1)$ (1 XOR + 1 AND + 1 OR gate) (hardware for condition evaluation) $+ k$ (1 XOR + 1 AND) gates (for error correction)
MBWing	(coding) $2mH_c$(Wing) (decoding) $2mH_d$(Wing) + global AND hardware
Interleaving scheme	(coding) $mH_c + n_1(m \times n_2\text{-}bit)$ circular shift registers (decoding) $mH_d + n_1(m \times n_2)$-bit circular shift registers + global AND hardware

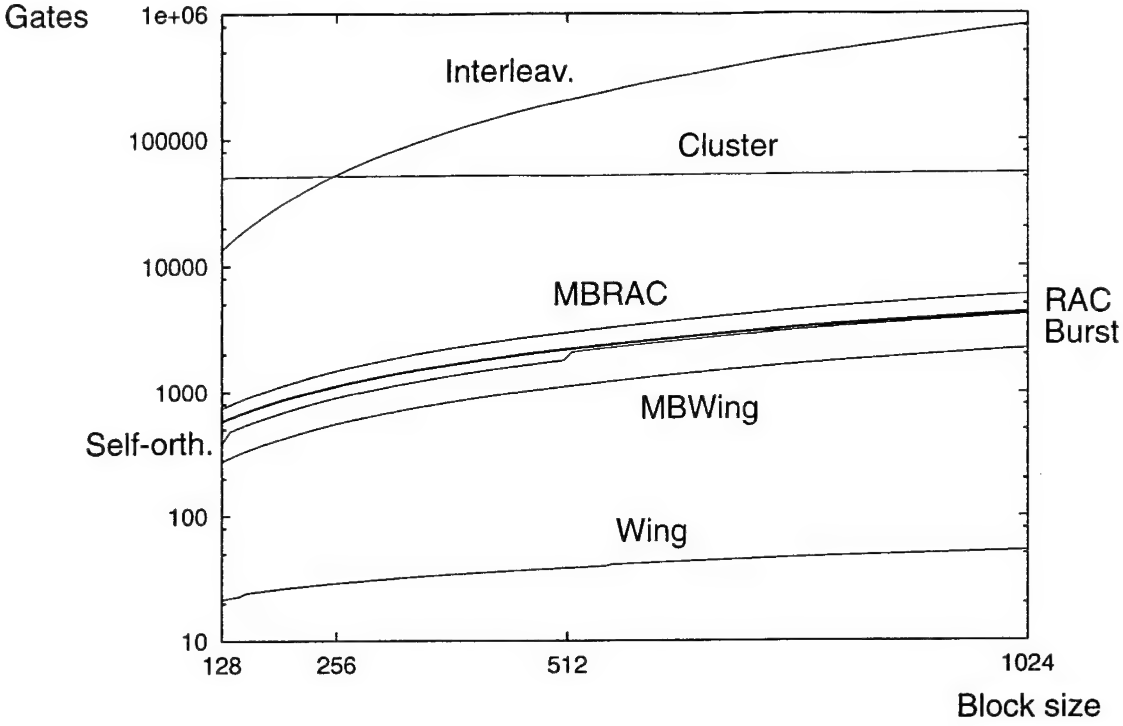


Figure 15. Hardware complexity (decoding, parallel implementation).

Table 8. Interconnection complexity (parallel implementation only)

Code	*Interconnection complexity*
RAC	$2n$
MBRAC	$2n + m \times \min(n_1, n_2)$
Self-orthogonal	(max) $p^2 \times \frac{p(p+1)}{2}$
Cluster	–
Single burst	–
Wing	$2k$
MBWing	$2mp'(p'+1) + m(p'+1)$
Interleaving scheme	$k'(m+2) + m$

Table 9. Coding delay

Code	*Coding delay (serial)*	*Coding delay (parallel)*
RAC	$\left[2k_2k_1{-}1 - \max(k_1, k_2)\right]T_{XOR}$	$(\lceil \log_2 k_1 \rceil + \lceil \log_2 k_2 \rceil)T_{XOR}$
MBRAC	$mT_{cd}(\text{RAC})$	$T_{cd}(\text{RAC})$
Self-orthogonal	$\left(p^2 + (\delta{-}1)p\right)T_{SHIFT}$	$\lceil \log_2 p \rceil T_{XOR}$
Cluster	$\left[2k_2k_1{-}1 - \max(k_1, k_2)\right]T_{XOR} +$ $+ \left[\frac{n_1}{b_1}\sum_{i=0}^{b_1-1} ib_2 + \frac{n_2}{b_2}\sum_{i=0}^{b_2-1} ib_1\right] T_{SHIFT}$	$(\lceil \log_2 k_1 \rceil + \lceil \log_2 k_2 \rceil)T_{XOR} +$ $+ \left((b_1 - 1)b_2 + b_1(b_2 - 1)\right)T_{SHIFT}$
Single burst	$\left[2k_2k_1{-}1 - \max(k_1, k_2)\right]T_{XOR} +$ $+ nT_{MEM}$	$(\lceil \log_2 k_1 \rceil + \lceil \log_2 k_2 \rceil)T_{XOR} +$ $+ T_{COMM}$
Wing	$2kT_{XOR}$	$\lceil \log_2(p - 1) \rceil T_{XOR}$
MBWing	$2mT_{cd}(\text{Wing})$	$T_{cd}(\text{Wing})$
Interleaving scheme	$mT_{cd}(\text{RAC}) + 2nT_{MEM}$	$T_{cd}(\text{RAC}) + (m - 1)T_{SHIFT}$

Table 10. Decoding delay (serial implementation)

Code	*Decoding delay*
RAC	$[\underbrace{2k_2k_1-1 - \max(k_1,k_2)}_{\textit{Check-bit computation}} + \underbrace{(k_1 + k_2 + 1)}_{\textit{Check-bit comparison}}]T_{XOR} + T_{COND\text{-}SER}$ where $T_{COND-SER} = (k_1 + k_2 - 2)(T_{XOR} + T_{AND} + T_{OR})$
MBRAC	(max) mT_{dd}(RAC)
Self-orthogonal	$\left(2p^2 + (\delta-1)p\right)T_{SHIFT}$
Cluster	Coding delay + (syndrome computation) $+ \left(k_1 + k_2 + 1\right) T_{XOR}$ (check for errors) $+ \left(k_1 + k_2\right) T_{OR}$ (if errors occurred: error bit counting) $+ \left(k_1 + k_2\right) T_{ADD}$ (if correctable errors: error correction) $+ O(b_1b_2)$ CPU operations
Single burst	$2nT_{SHIFT} + nT_{MEM}$
Wing	$(2k + p)T_{XOR}$ (condition evaluation) $+ (p - 1)\ (T_{XOR} + T_{AND} + T_{OR})$ (if correction) $+\ T_{XOR}$
MBWing	(max) $2mT_{dd}$(Wing)
Interleaving scheme	mT_{dd} (for condition evaluation) $+\ mT_{AND}$ (if correction) $+\ eT_{XOR}$

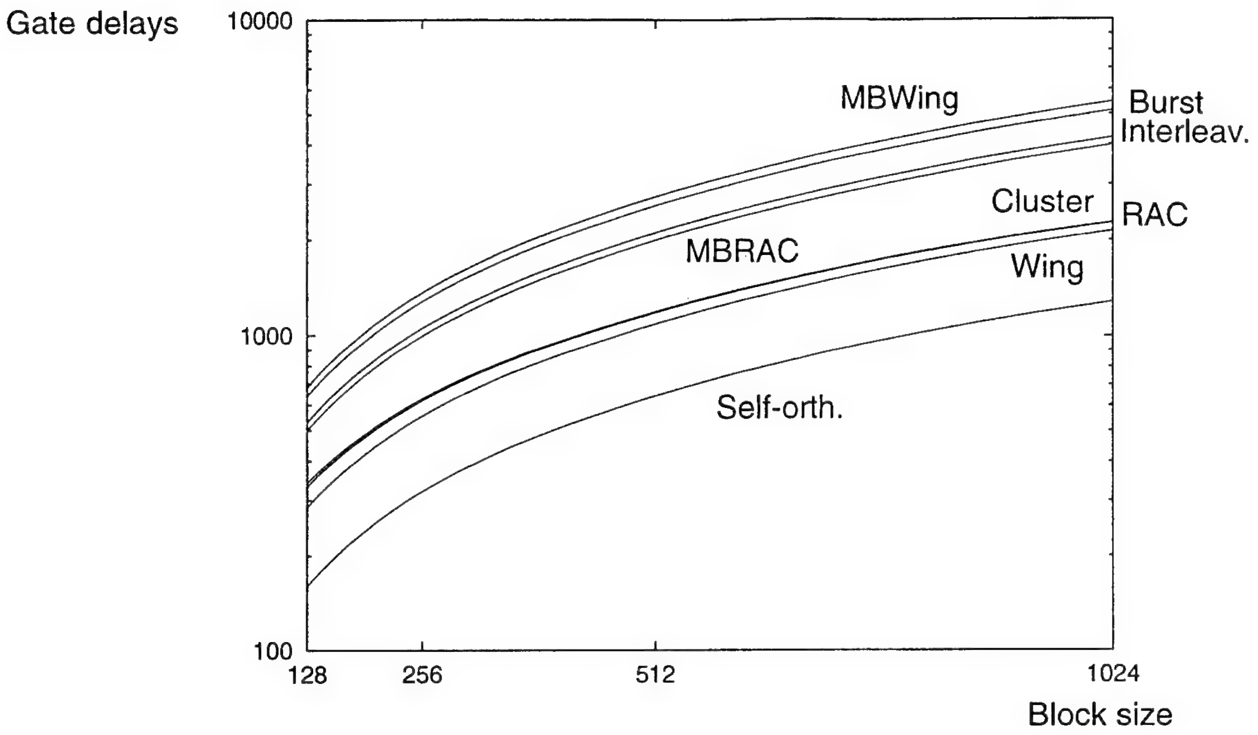


Figure 16. Decoding delay (serial implementations).

Table 11. Decoding delay (parallel implementation)

Code	*Decoding delay*
RAC	$(\lceil \log_2 k_1 \rceil + \lceil \log_2 k_2 \rceil + 1)T_{XOR} + \max(k_1, k_2)T_{XOR}$
MBRAC	T_{dd}(RAC) $+ T_{AND}$ (global AND)
Self-orthogonal	$(\lceil \log_2 p \rceil + 2)T_{XOR} + T_{MAJ} + T_{count}$
Cluster	Coding delay + (syndrome computation) $+ T_{XOR}$ (check for errors) $+ \max(\lceil \log_2 k_1 \rceil, \lceil \log_2 k_2 \rceil)\ T_{OR}$ (if errors occurred: error bit counting) $+ (k_1 + k_2)\ T_{ADD}$ (if correctable errors: error correction) $+ O(b_1 b_2)$ CPU operations
Single burst	Coding delay + (syndrome computation) $+ T_{XOR}$ (check for errors) $+ \lceil \log_2 k_2 \rceil\ T_{OR}$ (if errors occurred: error bit counting) $+ k_2\ T_{ADD}$ (if correctable errors: error correction) $+ k_1 T_{XOR}$
Wing	$(\lceil \log_2 (p - 1) \rceil + 1)T_{XOR}$ (for condition evaluation) $+ pT_{XOR}$ (for correction) $+ T_{XOR}$
MBWing	T_{dd}(Wing) $+ T_{AND}$ (global AND)
Interleaving scheme	$(m - 1)T_{SHIFT} + T_{dd}$(RAC) $+ T_{AND}$ (global AND)

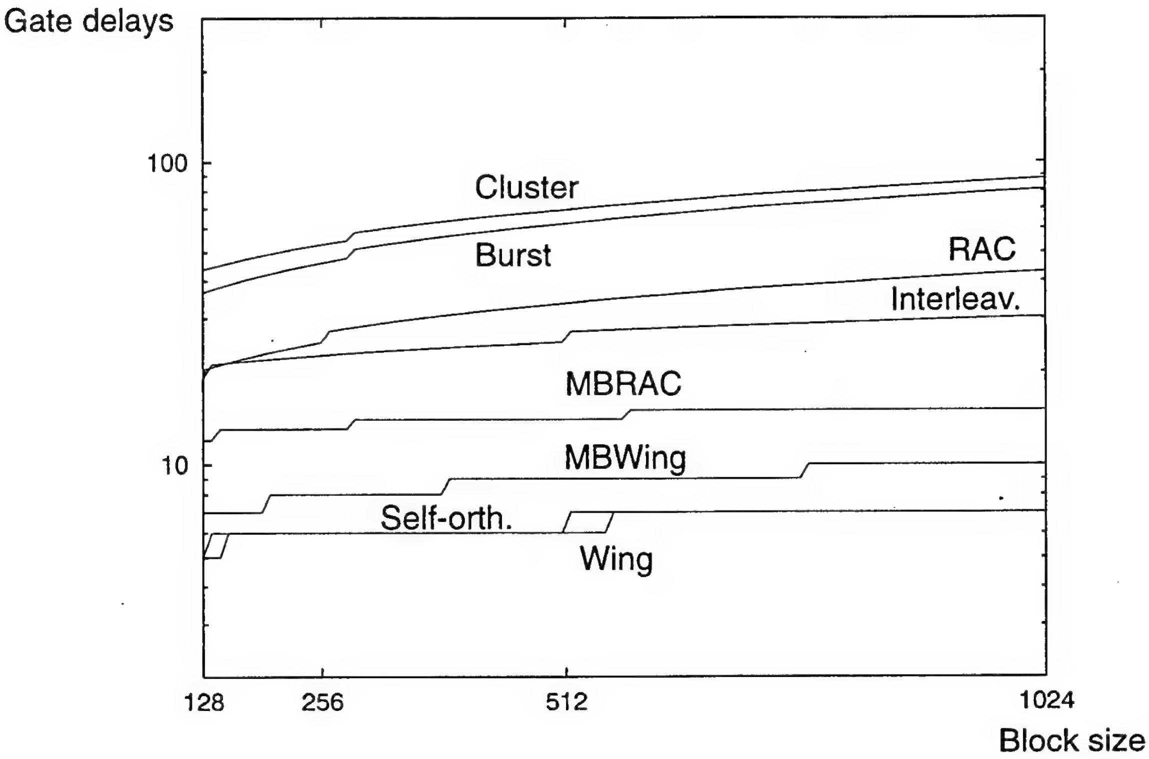


Figure 17. Decoding delay (parallel implementations).

4.5.2.3 Corrected bit-error rate

We have theoretically computed the corrected bit-error rate of the error correcting codes we have evaluated using the following upper bound,

$$P_d(P_e, n) \leq 1 - \left[\sum_{i=0}^{t}\binom{n}{i} P_e^i\ (1-P_e)^{n-i}\right]^{\frac{1}{n}}$$

As an example, for the RAC code this upper bound becomes,

$$P_d(P_e, n) \leq 1 - \left((1-P_e)^n - nP_e(1-P_e)^{n-1} - \frac{n(n-1)}{2}P_e^2(1-P_e)^{n-2}\right)^{\frac{1}{n}}$$

Note that for the RAC code $t = 2$, because this code is capable of detecting two errors

Next we plot $P_d(P_e, n)$ for the codes that we analyzed for raw bit-error rates ranging between 10^{-4} and 10^{-6} and appropriate block sizes. The corrected bit-error rate for a multiblock code is that of its component single block code.

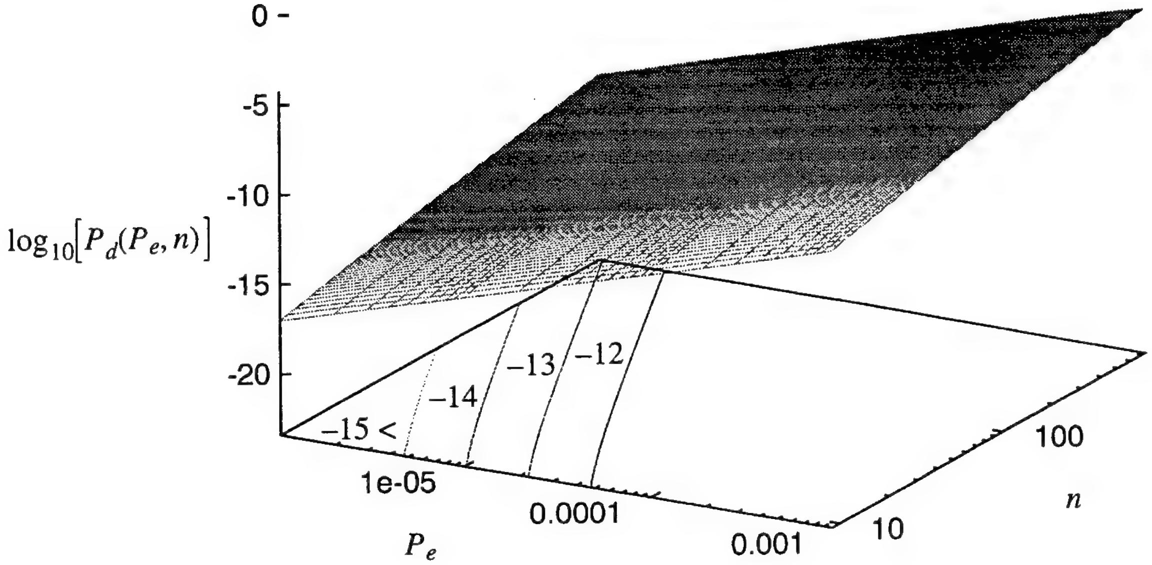


Figure 18. Plot of $\log_{10}[P_d(P_e, n)]$ for the RAC code.

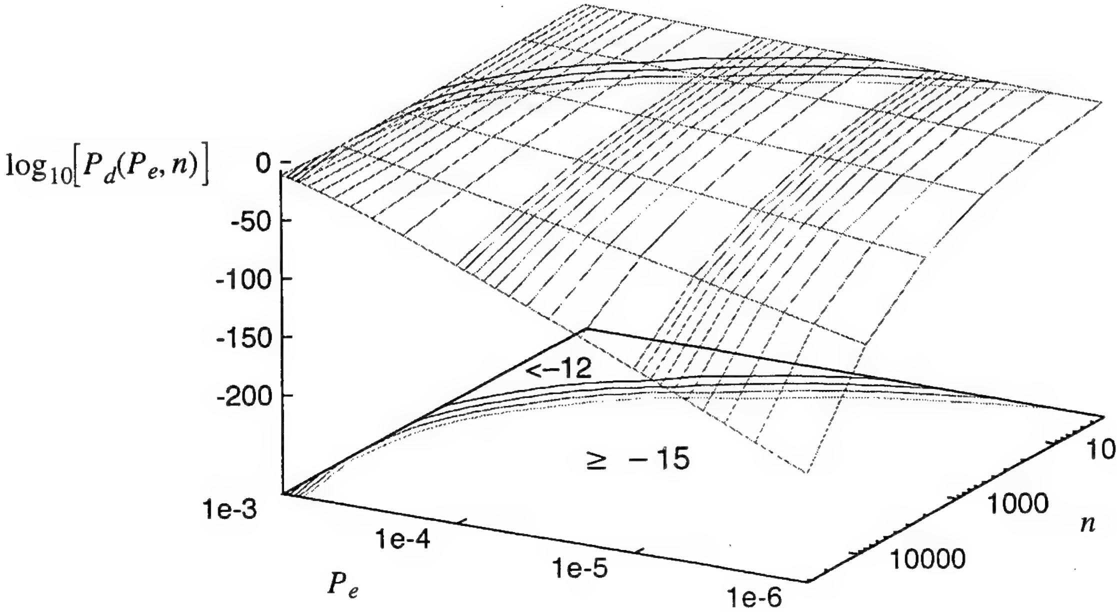


Figure 19. Plot of $\log_{10}[P_d(P_e, n)]$ for the self-orthogonal code with $r = 1/2$.

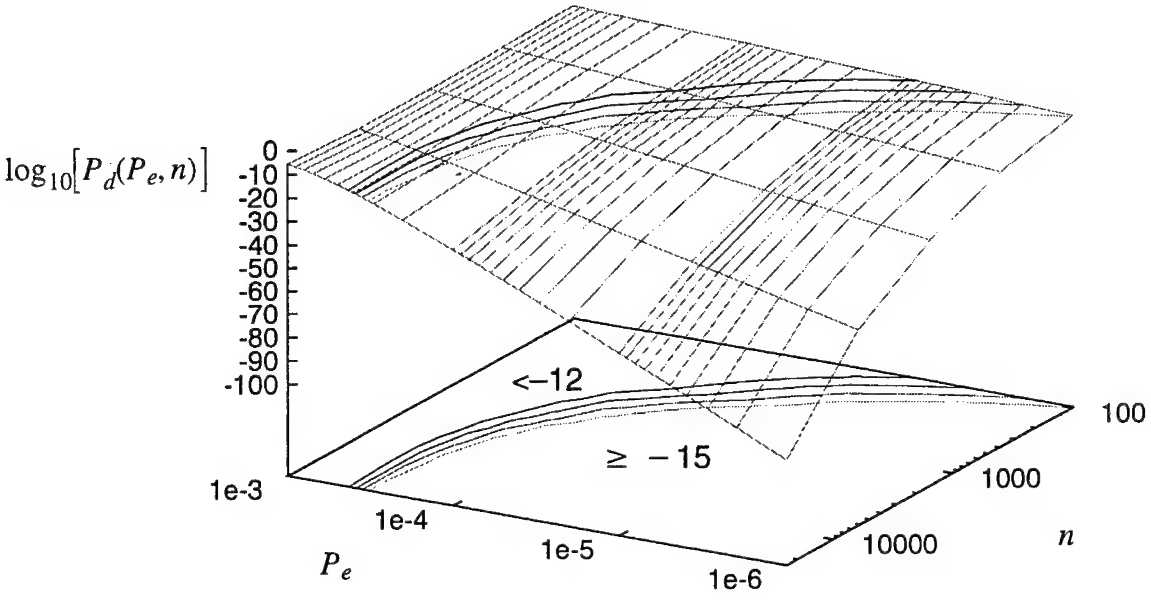


Figure 20. Plot of $\log_{10}[P_d(P_e, n)]$ for the self-orthogonal code with $r = 2/3$.

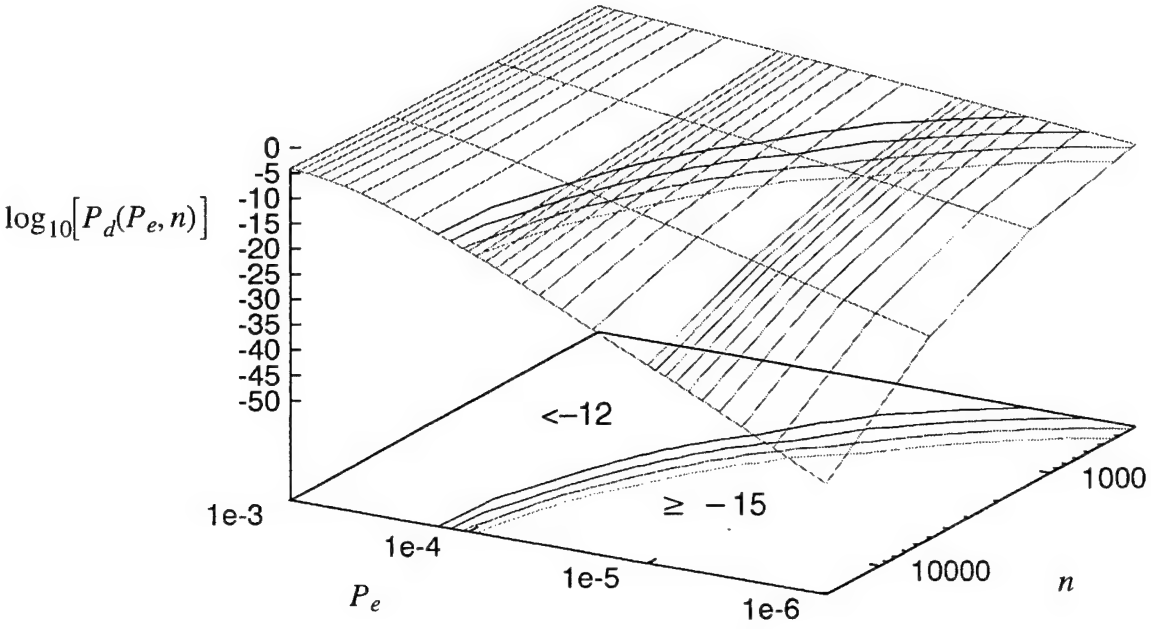


Figure 21. Plot of $\log_{10}[P_d(P_e, n)]$ for the self-orthogonal code with $r = 4/5$.

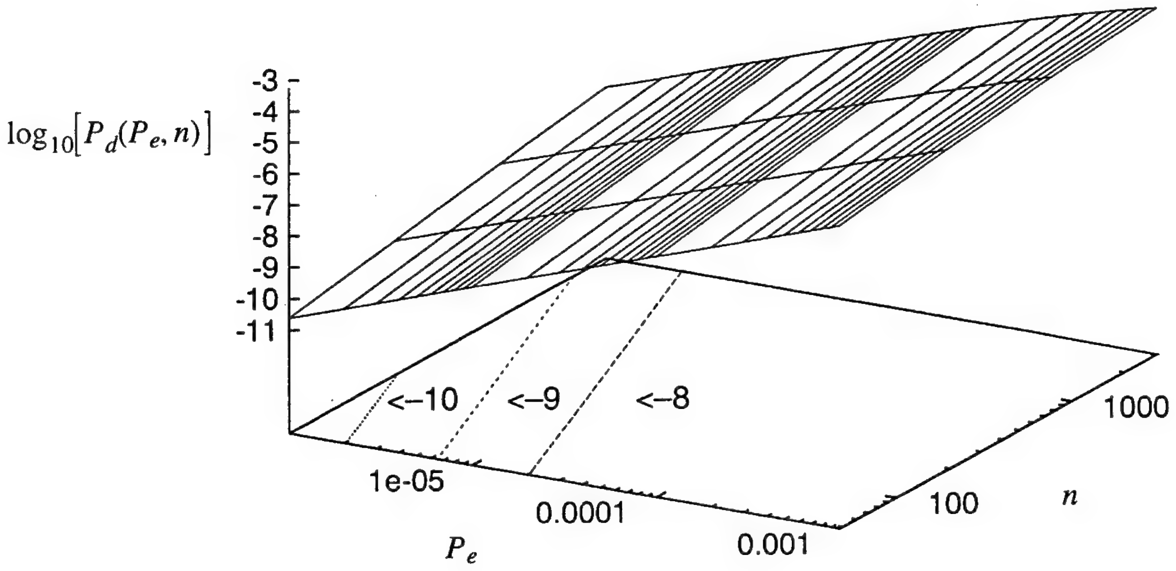


Figure 22. Plot of $\log_{10}[P_d(P_e, n)]$ for the 2 by 2 cluster-correcting code.

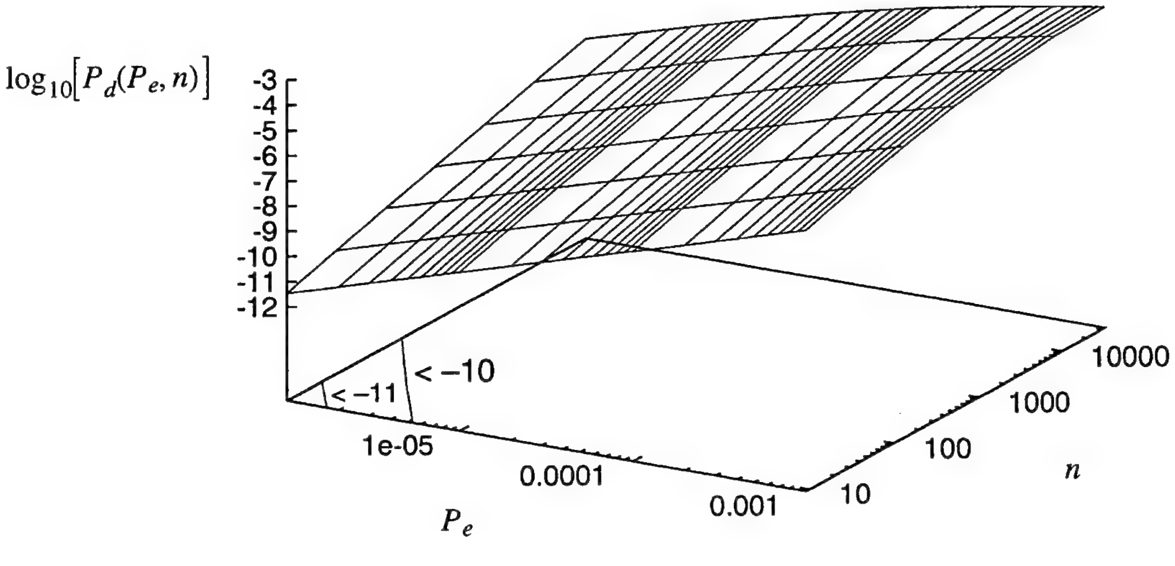


Figure 23. Plot of $\log_{10}[P_d(P_e, n)]$ for the single burst-correcting code.

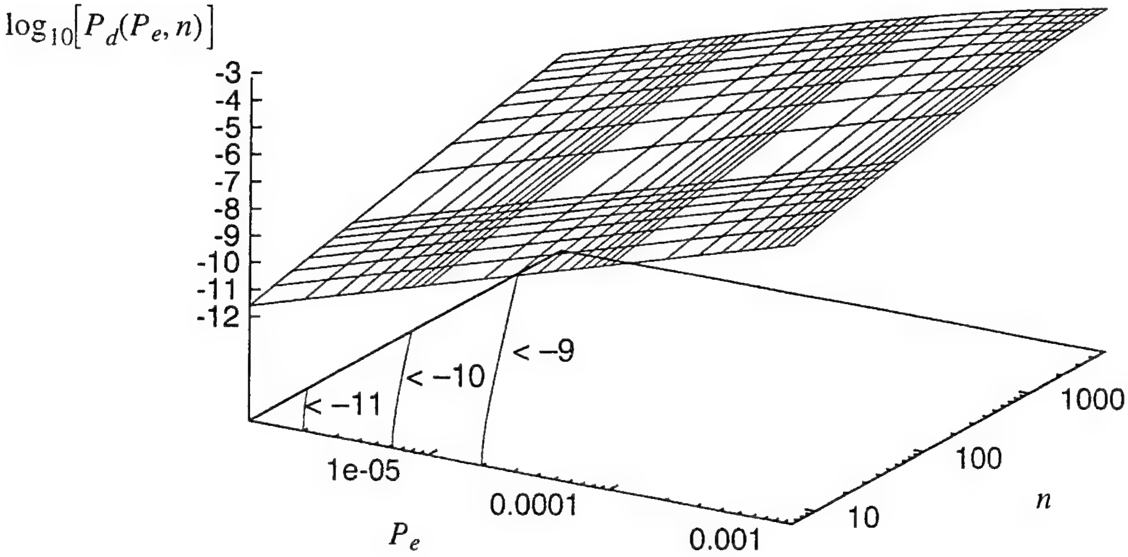


Figure 24. Plot of $\log_{10}[P_d(P_e, n)]$ for the wing code.

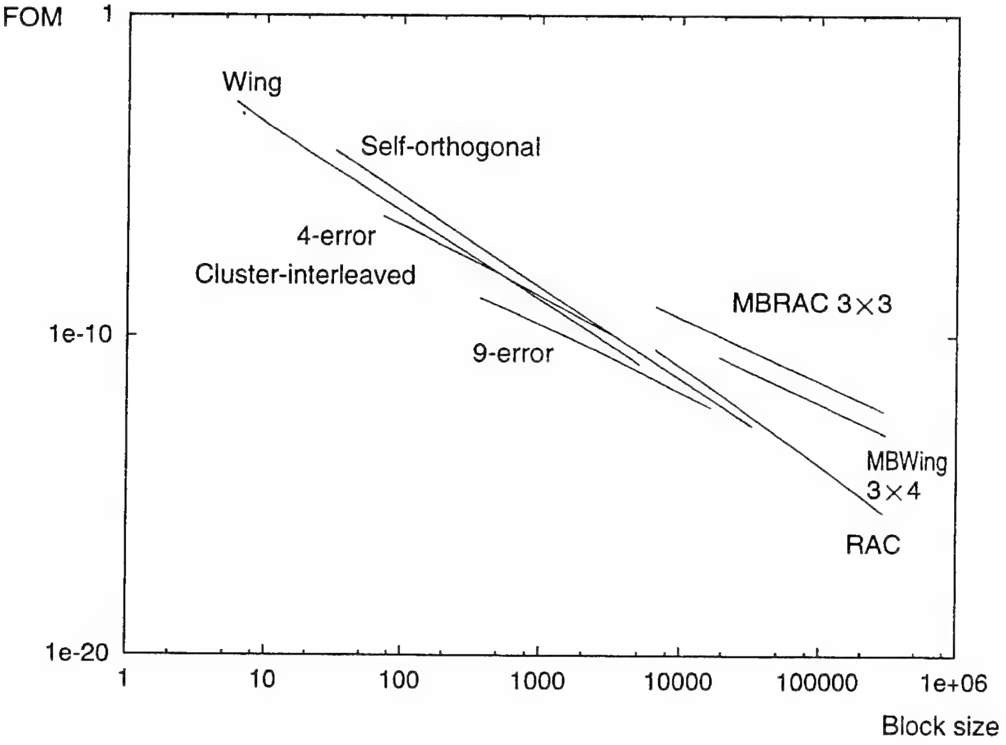


Figure 25. Comparison of bit-parallel implementations of self-orthogonal, cluster-interleaved, multiblock wing and RAC and single block wing and RAC codes, including the interconnection complexity.

4.5.2.4 Figure of merit

Figure 25 plots the FOM for some of the codes we have evaluated. While all the codes are presented on the same graph, the cluster correcting code is not intended for comparison with the random error correcting codes. All the implementations we compare are parallel. It is interesting to note that the wing and RAC codes attain some of the lowest values because they can only correct one error regardless of page size. The MBRAC code achieves much better values. The complex interconnections that the self-orthogonal code requires greatly reduce its FOM, even though this code can correct several errors. The cluster-interleaved code also requires complex interconnections.

4.6 Development of encoding schemes for page oriented memories

There are three different avenues for accomplishing this task: a) adoption of existing array codes with minor changes, if they are deemed suitable for POMs, b) modification of existing array codes, and c) development of new encoding schemes tailored to the specific needs of optical memories. We began with the first option and the results are included in the discussion of Section 4.5. Some work was performed along the lines of the second option and yielded the modified multiple block RAC code discussed in the bibliography reference [GOE96].

4.7 Investigation of optoelectronic techniques for parallel implementation of error detection and correction in POMs

We have begun investigating the feasibility of using "smart" photodetector arrays (SPAs) to interface the optical memory to an electronic host computer. SPAs have been suggested as a means to overcome the limitations of CCDs [1]. Here, an optical data page is received by two-dimensional arrays of "smart" photodetector elements, or pixels, replacing conventional CCDs. Embedded in each pixel is a simple processing element. The processing elements use local electronic interconnects to perform highly parallel, fine-grain computations. The SPA can be designed to perform fast parallel error control and data reduction, thereby providing a more efficient interface between the POM and the electronic computer. This architecture optimizes the computer memory system by combining the massive parallelism and high speed of optics with the diverse functionality, low cost, and local interconnection efficiency of electronics.

The time required to retrieve a data page from the POM and transfer it onto the photodetector array is, in general, independent of the page size. A higher throughput can be achieved with larger pages since for a page dimension, n, throughput increases with n^2 (assuming a square data page). It is useful, then, to know the scalability of a SPA design, or how large the data page can be made, so that throughput can be maximized.

4.7.1 SPA design example

SPAs can be implemented by integrating arrays of photodetectors with VLSI logic circuits thereby taking advantage of the speed, density, low power, and well-established fabrication processes of silicon CMOS VLSI. The possible functions of SPAs are, of course, unlimited. Here we provide a simple example of how SPAs can perform error detection and correction.

A*rray codes* are highly suited to the two-dimensional data format of POMs because these codes treat data as two-dimensional arrays of bits [2]. The simplest array code, the RAC code, generates parity bits along the rows and columns of an $I \times J$ bit data array and appends them to the original array. When data are read, the intersection of ones in the parity check vectors indicates the exact position of a single error anywhere in the $(I+1) \times (J+1)$ array. Since the cross parity code is a distance-4 code, it can be used in a single-error correcting-double-error detecting mode. To achieve a corrected bit error rate of 10^{-12} for a system raw bit error rate of 10^{-5}, code blocks can contain up to 80 bits. For large page sizes with uniformly distributed errors, the multiblock RAC code will be preferable (see also FOM discussion).

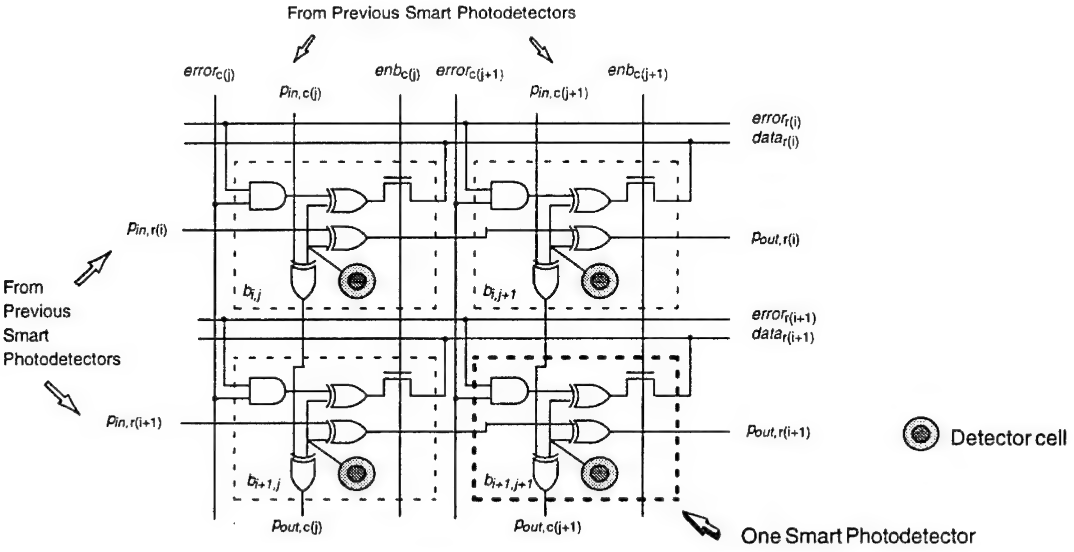


Figure 26. Smart photodetector array implementation of the RAC error correction code.

The circuit in Figure 26 demonstrates how a RAC code can be implemented in a SPA. The detector cell includes a photodetector and photoreceiver circuit to convert the optical energy into an electrical signal. The photoreceiver circuit can be fabricated on the same substrate as the SPA to take advantage of the low power and high density of CMOS VLSI. A simple receiver can be designed using only 2 transistors, however, more practical designs may require several (8–12) transistors. The photodetector element may also be integrated on the SPA substrate (e.g. a silicon photodiode), however because CMOS processing is not optimized for optical devices, better performance can be achieved by fabricating the detector separately and interfacing it to the SPA using hybrid techniques such as flip chip bump bonding and subsequent substrate removal [3]. This approach provides additional flexibility in detector selection since many detectors such as PIN diodes, HPTs and APDs are not readily integrated with CMOS VLSI processing. In addition, responsivity can be increased due to backside illumination of the flip chip bonded detector.

4.7.2 Design considerations

4.7.2.1 Hardware Complexity

The scalability of SPAs depends, in part, on the area of the individual pixels which comprise the array. The pixel area in turn depends on the complexity of the processing element. Because SPAs are used for fine-grain computations, local interconnections are assumed, and hardware complexity can be measured in terms of the number of gates in each processing element, or pixel. Here a gate is assumed to be a 2-input NAND equivalent.

Maximum page size is determined by yield-limited die size. We define *integration capability* as the average area per gate for a given fabrication technology. This empirically determined area estimate includes device area and average interconnect area for low complexity standard cell designs and can be used to conservatively model the fine-grain computing architecture of SPAs. For state-of-the-art VLSI processing (0.35 micron drawn minimum geometry), the integration capability is approximately 0.2 mil^2 per gate. Given the integration capability, maximum page size can then be computed as a function of smart pixel complexity. Figure 27a shows the maximum page size assuming a fixed detector size of 100 μm^2 and a practical maximum core chip area (excluding control circuitry and I/O pads) of 1 cm^2. The fixed detector area is assumed to be the active area of an on-chip photodetector and is comparable to CCD pixel size [4]. Figure 27b shows the maximum page size for SPAs for the case of hybrid integration. Using flip chip bump bonding, smart logic can be incorporated under the bump bond sites, resulting in increased scalability of the SPA.

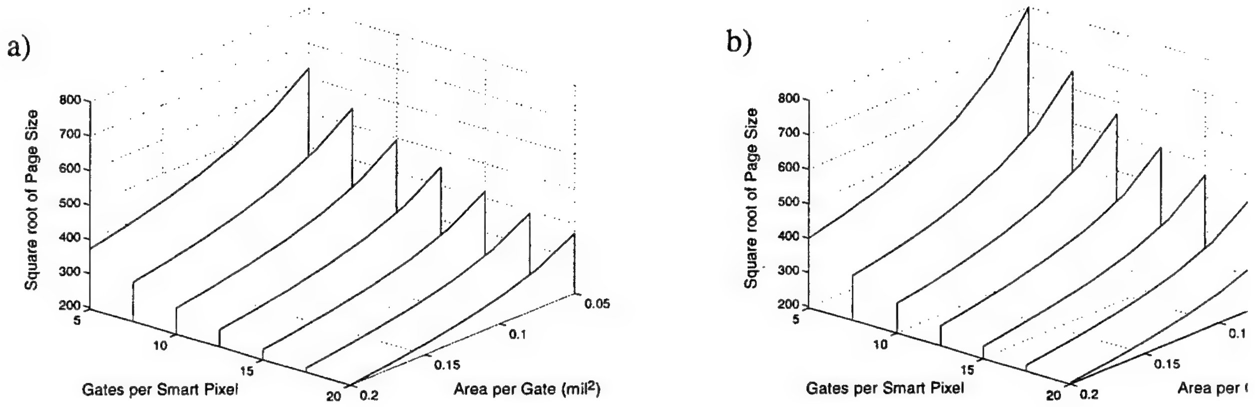


Figure 27. SPA page size vs. complexity vs. integration capability (SPA core area = 1 cm^2) a) SPA with 100 μm^2 detector, b) SPA with flip chip bonded detector.

Comparing Figures 27a and 27b, the on-chip detector has a small impact on SPA page size for state-of-the-art fabrication (i.e., Area per Gate = 0.2 mil^2). However, this difference in scalability increases greatly with further advances in VLSI processing (i.e. line width reductions) because the photodetector does not scale accordingly. The flip chip bonding approach shown in Figure 27b allows for SPA designs which take full advantage of VLSI integration capability. From Figure 27b, low complexity (5 gates per smart pixel) SPAs of size 512 × 512 can be achieved with the next generation CMOS technology (0.1 – 0.15 mil^2 per gate). This SPA complexity approximates the circuitry required for the RAC code in Figure 26. With two generations of CMOS line width reductions (0.05 – 0.1 mil^2 per gate), Figure 27b indicates that 512 × 512 SPAs having more complex SPA functionality, e.g. 10 gates per smart pixel, can be achieved.

SPA page sizes of 512 × 512 are desirable, since 2 × 2 mosaic arrays of independent SPA's can be combined to receive 1 Mbit data pages from an optical memory. This approach has been demonstrated for large-format CCD arrays, with each array being edge buttable on two sides. Using custom buttable packaging, "dead" space as small as 400 microns between CCD arrays has been achieved [5].

4.7.2.2 Optical Power

Optical efficiencies of POMs are extremely low, resulting in optical power to the SPA on the order of a few picowatts per pixel. Such low input power requires that noise sources be reduced to an absolute minimum. This is accomplished by selecting a detector with low dark current and minimizing thermal noise from receiver biasing resistors. The latter is achieved by an integrating, or high impedance, front end receiver [6]. The integration time, i.e. the time needed for the optically generated current to charge (or discharge) a MOS receiver gate to a voltage level sufficient to sense the optical signal, is given by $\tau_{\text{int}} = E_{sw} / P_{opt}$, where E_{sw} is the electrical device switching energy and P_{opt} is the input optical power. For state-of-the-art CMOS, devices can be designed with switching energies in the range of 50–100 fJ. This assumes 3V operation and some initial biasing so that full signal swing is not required at the receiver input. Figure 28a shows detector integration time vs. electrical switching energy for several different values of optical input power.

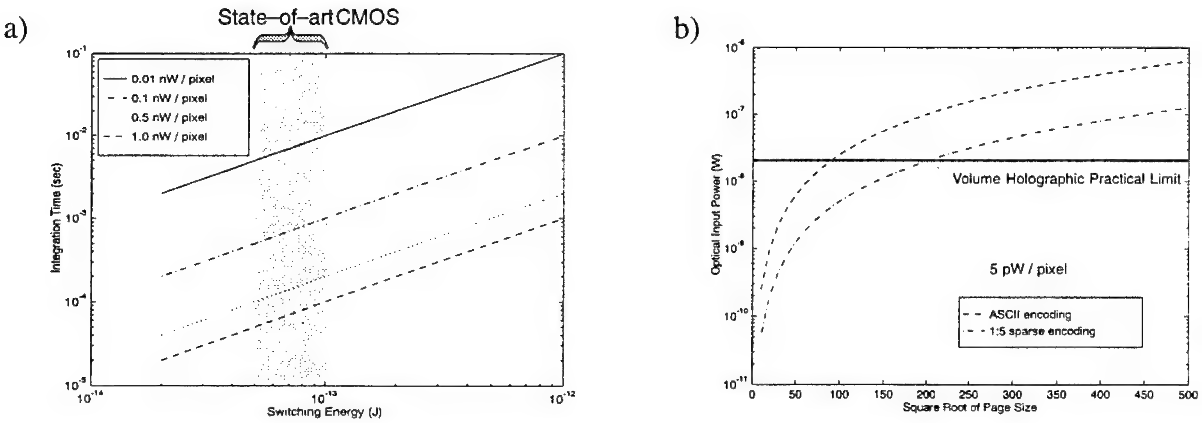


Figure 28. a) detector integration time vs. electrical switching energy
b) SPA input optical power vs. page size.

For detector integration to occur in 100 µsec, corresponding to a best case system clock rate of only 10 kHz, the required optical input power per pixel is approximately 0.5 nW which is two orders of magnitude higher than what is available from the POM. To achieve this equivalent power level, a detector with gain ≈ 100 is required. This amount of gain can be achieved with a phototransistor or avalanche photodiode, however, noise will also be amplified and thermal cooling of the SPA may be required. The low optical power budget limits the POM access time and reinforces the requirement for large page sizes to achieve high SPA throughput.

However, for a fixed input power level, larger page sizes also result in lower optical power per pixel, leading to an increase in integration time as Figure 28a demonstrates. Figure 28b plots the required optical input power versus page size, to achieve 5 pW per pixel. Assuming detector gain is 100, this power level will provide for integration in 100 µsec. For large page sizes, approximately 600 nW is required for ASCII encoding, corresponding to a code rate of 0.79 for the multiblock RAC error correction code (block size = 80). This level of optical input power presents a problem for volume holographic storage systems due to the extremely low diffraction efficiencies of these memories. Optical intensity is generally kept in the range of 15 mW/cm^2 to prevent writing during the read operation. For a 1 cm^3 photorefractive crystal with a diffraction efficiency of 10^{-6}, a relatively high 600 mW of incident optical power would be required, causing the crystal to saturate. The horizontal line in Figure 28b indicates the practical limit for volume holographic memories assuming a 1 cm^3 photorefractive crystal. The optical power budget does not limit page size for holographic storage applications in which the crystal has been fixed [8] thereby increasing the diffraction efficiency and/or the optical erase intensity. However, the memory is not readily written and effectively becomes a ROM.

The page size of a read/write holographic memory can be increased with sparse encoding [7], i.e. limiting the number of "on" bits. For a 1:5 sparse code, only one bit in a group of five bits is illuminated. This coding scheme helps to avoid saturating the crystal by reducing the required optical intensity, but results in a code rate of approximately 0.5 for the multiblock RAC error correction code. From Figure 28b, the 1:5 sparse encoding increases the page size for volume holographic memories from less than 100 × 100 pixels to 200 × 200 pixels. After accounting for the code rate reduction, the 1:5 sparse encoding more than doubles the amount of data per page.

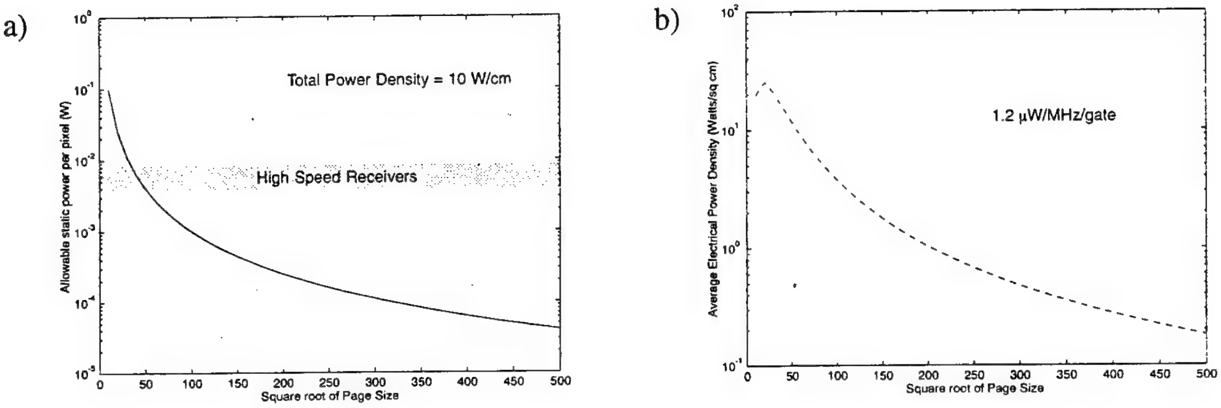


Figure 29. a) Static power per pixel vs. page size
b) dynamic power density vs. page size.

4.7.2.3 Electrical Power

The low optical power budget of POMs requires that SPAs incorporate sensitive receiver circuits. If an integrating receiver is used, some degree of biasing will likely be required to reduce detector integration time as previously discussed. This initial biasing will result in static power consumption. Feedback and gain stages in transimpedance receivers will also dissipate static power. Figure 29a shows the allowable static power per pixel assuming a maximum power density of 10 W/cm^2. Optical receivers operating up to 550 MHz have been designed with static power dissipation of 5 mW per receiver [9]. Such designs limit SPA page size to less than 50 × 50 pixels. However, the optical power budget limits detector integration time making such high speed receivers unnecessary. A 1 MHz receiver, for example, will only increase the signal conversion time by 1 μsec, or

1%. By leveraging this feature of POMs, slower receivers can be implemented, requiring less static power, and thereby allowing for larger page sizes.

Dynamic electrical power is considered for the complete smart photodetector circuit. This includes the SPA plus any additional addressing, sensing, control circuitry, etc. Figure 29b shows the dynamic electrical power density for a sample SPA circuit operating at 50 MHz assuming 1.2 μW/MHz/gate. Once the area of the smart pixels surpasses the area of the address and control circuitry, the dynamic power density drops with increasing page size. This is because the majority of the logic, i.e. the smart pixels, are only switching during the error detection/correction phase which occurs in a matter of a few clock cycles due to the high speed of VLSI electronics. As will be discussed in the next section, the majority of SPA operating time is spent integrating and reading out the SPA data, involving relatively little logic switching. As a result, page sizes greater than 100 × 100 can easily dissipate the dynamic electrical power.

4.7.3 SPA throughput

SPA throughput depends on three factors: 1) *signal conversion time,* 2) *SPA functional latency* and 3) *SPA read-out time.* Signal conversion time is the time required to integrate the optical signal and convert it to an acceptable electrical signal. It is on the order of 100 μsec for POMs as discussed in section 3.2. SPA functional latency is the time required for the SPA to perform its actual "smart" function, e.g. error correction, data compression, etc. Because SPAs use fine-grain computations, functional latency will require very few clock cycles and therefore is on the order of 10's to 100's of nanoseconds. Array read-out time is the time required to transfer the electronic data from the SPA to the host system. These three factors need not occur in sequence as pipeline operation is feasible.

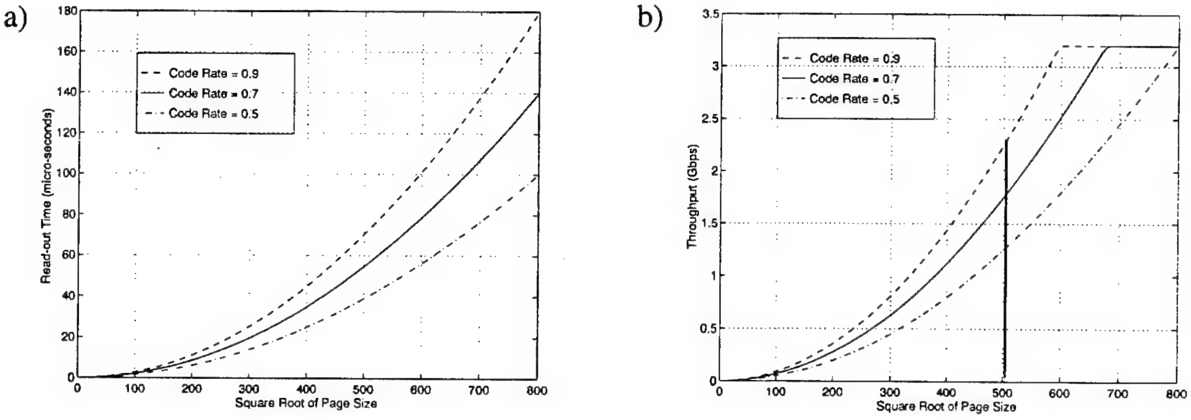


Figure 30. a) Array read-out time vs. page size
b) SPA throughput vs. page size for 100 μsec read-out time

SPA read-out time is, of course, design dependent. For our analysis a 64-bit output bus operating at 50 MHz is assumed. Array read-out time as a function of page size is plotted in Figure 30a for several code rate values. As code rate is increased, more clock cycles are required to output the additional information bits to the host system resulting in longer read-out times for a fixed page size. For page sizes smaller than 512 × 512 (limited by scalability factors previously discussed), read-out time is on the order of tens of microseconds – less than the signal conversion time, but greatly dominating the functional latency. To maximize overall SPA throughput, a sample and hold circuit can be implemented to integrate the next optical data page while the current page is processed and output to the host system.

Figure 30b shows total SPA throughput at several code rate values assuming sufficient detector gain to achieve 100 μsec optical integration. The array read-out clock speed can be relaxed so that the read-out time matches the integration time until the *saturation page size* is reached. The saturation page size occurs when the read-out time is 100 μsec for a 50 MHz clock rate and can be obtained from Figure 30a for each code rate. At the saturation page size, the maximum throughput for the bus structure, 3.2 Gbps, is achieved. For page sizes larger than the saturation value, the SPA read-out time is greater than 100 μsec and no further gains in throughput are realized. While it is desirable to maximize throughput by setting the SPA page size at the saturation value for a given error correction code, scalability constraints discussed previously require smaller page sizes. For the code rates shown, 1.3 – 2.3 Gbps *corrected* data rate can be achieved for SPA page size of 512 × 512. This is two orders of magnitude improvement over the *uncorrected* data rate of high speed frame-transfer CCDs [4].

4.7.4 Conclusions

We have examined several practical considerations related to the design of smart photodetector arrays for page-oriented memory interface applications. In order to maximize the high output rates of POMs, SPA page size must be large. State-of-the-art VLSI integration capabilities limit SPA size to approximately 350 × 350 for on-chip detectors and 400 × 400 for flip-chip bonded detectors. This difference will be further exaggerated with subsequent advances in VLSI integration. The low optical power budget of volume holographic memories greatly limits SPA page size. Innovative coding techniques such as sparse encoding can help alleviate this problem. In general, the low optical output power of POMs requires low dark current detectors with gain and an integrating front end receiver to minimize thermal noise. Integration time is on the order of 100 μsec due to the low optical power, so high performance receivers are not necessary. By implementing slower, low power receiver circuits, static current will not limit page size. SPA throughput is then limited by integration time, and a sample and hold circuit should be utilized to maximize throughput. Corrected data rates of 1.3–2.3 Gbps can be achieved for array sizes of 512 × 512. This yields two orders of magnitude improvement over the uncorrected data rate achievable with high speed frame transfer CCDs.

4.7.5 References

[1] M. E. Schaffer and P. A. Mitkas, "Smart photodetector arrays in silicon CMOS: practical considerations," in *Workshop on Data Encoding for Page-oriented Optical Memories*, Phoenix, Arizona, March 1996.

[2] P. G. Farrell, "Array Codes," in *Algebraic Coding Theory*, ed. by G. Longo, pp. 231–242, Springer-Verlag, 1979.

[3] K. W. Goossen et al., "GaAs MQW modulators integrated with silicon CMOS," *IEEE Photonics Technology Letters*, Vol. 7, No. 4, pp. 360–362, 1995.

[4] G. Williams and J. Janesick, "Cameras with CCDs capture new markets," *Laser Focus World*, Detector Handbook, pp S5–S9, 1996.

[5] J. C. Geary, G. A. Luppino, R. Bredthauer, R. J. Hlivak and L. Robinson, "A 4096 × 4096 pixel CCD mosaic imager for astronomical applications," in *Charge Coupled Devices and Solid State Optical Sensors II*, SPIE Vol. 1447, pp. 264–273, 1991.

[6] R. G. Smith and S. D. Personick, "Receiver Design for Optical Fiber Communication Systems," in *Topics in Applied Physics: Semiconductor Devices for Optical Communication*, ed. by H. Kressel, Vol. 39, Springer-Verlag, New York, 1982.

[7] B. J. Goertzen and P. A. Mitkas, "Error–correcting code for volume holographic storage of a relational database," *Optics Letters*, Vol. 20, No. 15, pp. 1655–1657, 1995.

[8] J. J. Amodei and D. L. Staebler, "Holographic pattern fixing in electro-optic crystals," *Applied Physics Letters*, Vol. 18, No. 12, pp. 540–542, 1971.

[9] A. V. Krishnamoorthy, et al., "Operation of a single-ended 550 Mbit/s, 41 fJ hybrid CMOS/MQW receiver-transmitter," *Electronics Letters*, Vol. 32, No. 8, pp.764–766, 1996.

4.8 Continuous determination of the status of the effort through regular progress reports

The continuous determination of the status of the effort was performed through regularly submitted progress reports. During the course of this project we provided 15 progress reports.

4.9 Final report

This document constitutes the final report of the project.

4.10 Oral presentations at such times and places as designated by the Rome Laboratory personnel

We have provided oral presentations as requested.

4.11 Future recommended research

Although we evaluated a few array codes, with the underlying goal of finding a code that can provide an acceptable CBER and that can be implemented in parallel with simple hardware, as Table 2 indicates, there are several other error correcting codes that should be considered for possible POM implementation. As a general rule, the more powerful the code, the more complex the hardware it would require for encoding and decoding a page of bits. This trade-off, will play a very significant role towards the production of an efficient and cost-effective POM system.

Characterization of a POM system channel is far from complete. Although some of the components of a volume holographic memory system have been well studied, there is a significant amount of work that needs to be done to fully understand the effects produced by some components and the interaction of all the components in the system. A good modeling of the POM system channel will greatly facilitate the task of selecting or designing an error correcting code for the system.

The simulator is a very useful tool that can be used for testing error codes and simulating real POM systems. It can greatly facilitate the channel modeling task and the testing of error codes. However, our simulator optimization work, in which we try to find the best possible values for the parameters of various error processes, so that the simulated results best match the real data, is not complete yet. Our approach of grouping the error processes into parallel blocks which are then applied in serial fashion to the input data, and finding values for their parameters using traditional system optimization techniques, has just been initiated. Further work is needed to produce a usable tool.

Finally, implementation of SPAs is very important and further work towards that goal needs to be done. SPAs can provide the throughput needed for a high-performance system. A test chip with an array of photodetectors and a simple error correcting code, such as the multiblock RAC code, embedded, is currently feasible and we can readily incorporate it to our POM system for testing purposes.

5. Bibliography

5.1 Error Coding

[1] [ABU88] Y. Abu–Mostafa (ed.), "Complexity in information theory", Springer Vale, NY, 1988.

[2] [ARG95] C. Argon and H. F. Ahmad, "Optimal optical orthogonal code design using difference sets and projective geometry," *Optics Communication*, Vol. 188, pp. 505–8, Aug 1995.

[3] [BAT89] S. D. Bate, B. Honary, and P. G. Farrell, "Error control techniques applicable to hf channels," *IEE Proc I*, Vol. 136, No. 1, pp. 57–63, Feb 1989.

[4] [BER83] E. R. Berlekamp, Algebraic coding theory, McGraw–Hill, NY, 1967. Note: Reprinted in Laguna Hills, CA by Algean Park Press, 1983.

[5] [BER80] E. R. Berlekamp, "The technology of error–correcting codes," *Proc. IEEE*, Vol. 68, No. 5, pp. 564–93, May 1980.

[6] [BLA96] M. Blaum, P. Farrell, and H. C. A. van Tilborg, "Array codes", submitted to *Handbook of Coding Theory*, eds. R. A. Brualdi, W. C. Huffman, and V. S. Pless, to be published by Elsevier.

[7] [BLA94] M. Blaum and P. G. Farrell, "Array codes for cluster–error correction," *Electronics Letters*, Vol. 30, No. 21, pp. 1752–3, Oct 13 1994.

[8] [BLA93] M. Blaum, R. Roth, "New array codes for multiple phased burst correction", *IEEE Transactions on Information Theory*, Vol. 39, No. 1, Jan 1993.

[9] [BLO89] A. A. Block, "Effect of data coding methods in holographic memory on characteristics of reconstructed images of data pages," *Optoelectron. Instrum. Data Process.*, Vol. 5, pp. 45051, 1989

[10] [BOS60] R. C. Bose and D. K. Ray–Chaudhuri, "On a class of error correcting binary group codes," in *Information and Control*, Vol. 3, pp. 68, 1960.

[11] [CHI69] R. T. Chien, "Burst–correcting codes with high–speed decoding," *IEEE Transactions on Information Theory*, Vol. IT–15, pp. 109–113, Jan 1969.

[12] [DOM89] S. A. Dombrovskii, "Effectiveness of using error–correcting codes in holographic storage systems," *Optoelectron. Instrumen.*, Vol. 2, pp. 58–62, 1989.

[13] [DOM88] V. A. Dombrovskii, S. A. Dombrovskii, and E. F. Pen, "Reliability of data readout in a holographic channel with constant characteristics," *Optoelectron. Instrumen. Data Process.*, Vol. 6, pp. 69–77, 1988.

[14] [FAR95] P. G. Farrell, "On generalized array codes", *to be published in the Proceedings of the Third International Symposium on Communication Theory and Applications*, Ambleide, UK, July, 1995.

[15] [FAR95] P. G. Farrell, "Array codes: a tutorial summary", *to be published in DEPOMS handbook*, Phoenix, AZ, March 1996.

[16] [FAR92] P. G. Farrell, "A survey of error control codes," *European Journal on Telecommunications*, Vol. 3, No. 5, pp. 441–53.

[17] [FAR90] P. G. Farrell, "Coding as a cure for communication calamities," *Electronics and Communication Engineering Journal*, Vol. V2, No. 6, pp. 23–20, Dec 1990.

[18] [FAR89] P. G. Farrell, "An introduction to array codes," *Advanced School on Geometries*, 1989, Note: To be published by Springer–Verlag.

[19] [FAR85] P. G. Farrell, "Code structure and decoding complexity," in *The Impact of Processing Techniques on Communications*, J. K. Skwirzynski (ed.), Nijhoff, NY, 1985.

[20] [FIR59] P. Fire, "A class of multiple–error–correcting codes for non–independent errors," *Sylvania Reconnaissance Systems Lab Report*, No. RSL–E–2, 1959.

[21] [GAB91] E. Gabidulin, "A fast matrix decoding algorithm for rank–error–correcting codes", in G. Cohen, S. Litsyn, A. Lobstein, and G. Zémor editors of *Algebraic Coding: First French–Soviet Workshop*, Paris: Springer–Verlag, July 22–24, 1991.

[22] [GAL68] R. G. Gallager, "Information theory and reliable communications," Wiley, NY, 1968.

[23] [GOE95] B. J. Goertzen and P .A. Mitkas, "Error–correcting code for volume holographic storage of a relational database," *Optics Letters*, Vol. 20, No. 15, pp. 1655–7, Aug 1 1995.

[24] [GOE96] B. J. Goertzen and P .A. Mitkas, "Volume holographic storage for large relational databases," *Optical Engineering*, Vol. 35, No. 7, pp. 1847–1853, July 1996.

[25] [GOL49] M. J. E. Golay, "Notes on digital coding," *Proceedings of the IEEE (Correspondence)*, Vol. 37, pp. 657, 1949.

[26] [GOO93] R. M. Goodman, R. J. McEliece, and Masahiro Sayno, "Phased burst error–correcting array codes," *IEEE Transactions on Information Theory*, Vol 39, No. 2, pp. 684, Mar 1993.

[27] [GOO90] R. M. Goodman and M. Sayano, "Size limits on phased burst error correcting array codes," *Electronics Letters*, Vol. 26, No. 1, pp. 55–6, Jan 4 1990.

[28] [GOO75] R. M. F. Goodman and P. G. Farrell, "Data transmission with variable redundancy error control over a high–frequency channel," *Proceedings of the IEE*, Vol. 122, No. 2, pp. 113–8, Feb 1975.

[29] [GOP70] V. D. Goppa, "A new class of linear correcting codes," *Problems of Information Transmission*, Vol. 6, pp. 24, 1970.

[30] [HAM50] R. W. Hamming, "Error detecting and correcting codes," *Bell Systems Technical Journals*, Vol. 29, pp. 147, 1950.

[31] [HEA95a] J. F. Heanue, M. C. Bashaw, L. Hesselink, "Channel codes for digital holographic data storage," *Journal of the Optical Society of America A*, Vol. 12, No. 11, pp. 2432–9, Nov 1995.

[32] [HEN94] W. Henkel and H. Y. Chung, "New filling procedure to reduce delay of burst–error correcting array codes," *Electronics Letters*, Vol. 30, No. 6, pp. 465, Mar 17, 1994.

[33] [HIL86] R. Hill, "A first course in coding theory," NY: Clarendon: 1986.

[34] [HON95] B. Honary, G. Markarian, M. Darnel, "Low–complexity trellis decoding of linear block codes," *IEE Proc.–Commun.*, Vol. 142, No. 4, pp. 201, Aug 1995.

[35] [HON93] B. Honary, G. S. Markarian, and P. G. Farrell, "Generalized array codes and their trellis structure," *Electronics Letters*, Vol. 29, No. 6, pp. 541, Mar 18, 1993.

[36] [INO90] T. Inoue, H. Yoshida, and Y. Ishida, "The superimposed codes and its application to digital VCR," in *1990 International Symposium on Information Theory and Its Applications (ISITA '90)*, Hawaii, Nov 27–30, 1990.

[37] [INO85] T. Inoue, and others, "New encoding and decoding methods for generalized product codes," in *4th IEEE International Symposium on Informational Theory*, Brighton, June 1985.

[38] [MAC77] F. J. MacWilliams and N. J. A. Sloane, "The theory of error correcting codes," North Holland Publishing, NY, 1977.

[39] [NEI95a] M. A. Neifeld and J. D. Hayes, "Error–correction schemes for volume optical memories", *Applied Optics*, Vol. 34, No. 35, pp. 8183–90, Dec 10 1995.

[40] [NEI94a] M. A. Neifeld and M. McDonald, "Error correction for increasing the usable capacity of photorefractive memories," *Optics Letters*, Vol. 19, No. 18, pp. 1483–5, Sept 15 1994.

[41] [NEI94b] M. A. Neifeld and J. D. Hayes, "Parallel error correction for optical memories," *Optical Memory and Neural Networks*, Vol. 3, No. 2, pp. 87–98, 1994.

[42] [NG78] W. H. Ng and R. M. F. Goodman. "An efficient minimum distance decoding algorithm for convolutional error–correcting codes," *Proceedings of the IEE*, Vol. 125, No. 2, pp. 97–103, 1978.

[43] [OLS94] B. H. Olson and S. C. Esener, "Partial response precoding for parallel–readout memories," *Optical Letters*, Vol. 19, No. 4, pp. 661–3, May 1 1994.

[44] [PET72] W. W. Peterson and E. J. Weldon Jr., "Error correcting codes," Second Edition. NY: Massachusetts Institute of Technology: 1972.

[45] [RAO89] T. R. N. Rao and E. Fujiwara, "Error–control coding for computer systems," Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[46] [SMI77] R. J. G. Smith, "Easily–decoded efficient self-orthogonal block codes," *Electronics Letters*, Vol. 13, No. 7, March 31 1977.

[47] [TOW67] R. L. Townsend and E. J. Weldon Jr, "Self–orthogonal quasi–codes," *IEEE Transactions on Information Theory*, Vol. IT–13, pp. 390–4, 1967.

[48] [UNG78] G. Ungerboeck, "Trellis–coded modulation with redundant signal sets: Pt. 1: Introduction, Pt. 2: State of the art," *IEEE Communications Magazine*, Vol. 25, No. 2, pp. 5–21, Feb 1978.

[49] [XIA91] P. Xiao–Hong and P. G. Farrell, "Self–orthogonal codes constructed from cyclic squares", in *6th International Conference on Digital Processing of Signals in Communications*, Loughborough, UK, Sept 2–6 1991, IEE, London, pp. 244–5, 1991. Notes: Conference publication #340.

5.2 Page-Oriented Optical Memory Systems

[50] [ALT95] S. B. Altner, S. Bernet, A. Renn, E. S. Maniloff, F. R. Graf, and U. P. Wild, "Spectral holeburning and holography VI: photon echoes from CW spectrally programmed holograms in a Pr^{3+}:Y_2SiO_5 crystal," *Optical Communication*, Vol. 120, No. 1&2, pp. 103–12, Oct 1995.

[51] [AN95] X. An and D. Psaltis, "Experimental characterization of an angle multiplexed holographic memory," *Optics Letters*, Vol. 20, No. 18, pp. 1913–15, Sept 15 1995.

[52] [AND94] P. E. Andersen and A. Marrakchi, "Noise–free holographic storage in iron–doped lithium niobate crystals," *Optics Letters*, Vol. 19, No. 19, pp. 1583–5, Oct 1 1994.

[53] [BAB94] W. R. Babbitt and T. W. Mossberg, "Quasi–two–dimensional time–domain color memories: process limitations and potentials," *Journal of the Optical Society of America B*, Vol. 11, No. 10, pp. 1948–53, Oct 1994.

[54] [BAR95a] S. Bartkiewicz and A. Miniewicz, "Methylene blue sensitized poly(methyl methacrylate) matrix: a novel holographic material," *Applied Optics*, Vol. 34, No. 23, pp. 5175–8, Aug 10 1995.

[55] [BAR95b] M. Barikani, E. Simova, and M. Kavehrad, "Dichromated polyvinyl alcohol as a real–time hologram recording material: some observations and discussions," *Applied Optics*, Vol. 34, No. 13, pp. 2172–9, May 1 1995.

[56] [BER95] S. Bernet, S. B. Altner, F. R. Graf, E. S. Maniloff, A. Renn, and U. P. Wild, "Frequency and phase swept holograms in spectral hole–burning materials," *Applied Optics*, Vol. 34, No. 22, pp. 4674–84, Aug 1 1995.

[57] [BRA92] D. Brady and D. Psaltis, "Control of volume holograms," *Journal of the Optical Society of America A*, Vol. 9, pp. 1167–70, 1992.

[58] [BUR94] G. W. Burr, F. H. Mok, and D. Psaltis, "Large scale volume holographic storage in the long interaction length architecture," in *39th SPIE International Symposium on Optical Applied Science and Engineering Conference on Very Large Optical Memories*, San Diego, CA, July 1994.

[59] [BUR93a] G. W. Burr, "Holographic random access memory," *Optical Processing and Computing*, in SPIE Newsletter, Vol. 4, No. 2, pp. 5, Nov 1993.

[60] [BUR93b] G. W. Burr, K. Curtis, and D. Psaltis, "Comparison of wavelength– and angle–multiplexed holographic memories," in *OSA Topical Meeting on Optical Computing*, paper OWB4, March 1993.

[61] [CAM96] S. Campbell, Y. Zhang, P. Yeh, "Writing and copying in volume holographic memories: approaches and analysis", *Optics Communication*, Vol. 123, pp. 27–33, Jan 15 1996.

[62] [CAM95] S. Campbell, P. Yeh, C. Gu, S. H. Lin, C. Cheng, and K. Y. Hsu, "Optical restoration of photorefractive holograms through self–enhanced diffraction," *Optics Letters*, Vol. 20, No. 3, pp. 330–3, Feb 1 1995.

[63] [CAM94] S. Campbell, X. Yi, and P. Yeh, "Hybrid sparse–wavelength angle–multiplexed optical data storage systems," *Optics Letters*, Vol. 19, No. 24, pp. 2161–4, Dec 15, 1994.

[64] [CAR83] N. W. Carlson, L. J. Rothberg, and A. G. Yodh, "Storage and Time Reversal of Light Pulses using Photon Echoes," *Optics Letters*, Vol. 8, pp. 483+, 1983.

[65] [CHE75] D. Chen and J. D. Zook, "An overview of optical data storage technology," *Proceedings of the IEEE*, Vol. 63, pp. 1207+, 1975.

[66] [CUR94a] K. Curtis, A. Pu, and D. Psaltis, "Method for holographic storage using peristrophic multiplexing," *Optics Letters*, Vol. 19, No. 13, pp. 993–4, July 1 1994.

[67] [DUE96] M. Duelli, R. S. Cudney, P. Günter, "Discrimination of enclosed images by weighted storage in an optical associative memory", *Optics Communication*, Vol. 123, pp. 49–54, Jan 14 1996.

[68] [DVO94] A. S. Dvornikov, S. E. Esener, and P. M. Rentzepis, "Three–dimensional optical storage memory by means of two–photon interaction," in *Optical Computing Hardware*, S. H. Lee (ed.), Academic Press Inc, NY, 1994, Chapter 11

[69] [ELL95] S. A. Ellett, T. F. Krile, and J. F. Walkup, "Throughput analysis of digital partitioning with error–correcting codes for optical matrix–vector processors," *Applied Optics*, Vol. 34, No. 29, pp. 6744–51, Oct 10 1995.

[70] [ESE92] S. Esener, Y. Fainman, J. Ford, and S. Hunter, "Two–photon three–dimensional memory hierarchy," in *Photonic Neural Networks*, SPIE, 1992, Vol. 1776, pp. 346–55.

[71] [ESE91] S. Esener and P .M. Rentzepis, "Two–photon 3–D optical memories," in *Optical Data Storage Technical Digest Series*, Optical Society of America A, Vol. 5, pp. 70–4, 1991.

[72] [HEA95a] J. F. Heanue, M. C. Bradshaw, and L. Hesselink, "Encrypted holographic data storage based on orthogonal–phase–code multiplexing," *Applied Optics*, Vol. 34, No. 36, pp. 6012–15, Sept 10 1995.

[73] [HEA95b] J. F. Heanue, M. C. Bashaw, and L. Hesselink, "Sparse selection of reference beams for wavelength and angular multiplexed volume holography," *Journal of the Optical Society of America A*, Vol. 12, No. 8, pp. 1671–6, Aug 1995.

[74] [HEA94a] J. F. Heanue, M. C. Bashaw, and L. Hesselink, "Volume holographic storage and retrieval of data," *Science*, Vol. 265, pp. 749–52, Aug 5 1994.

[75] [HEA94b] J. F. Heanue, M. C. Bashaw, and L. Hesselink, "Recall of linear combinations of stored data pages based on phase–code multiplexing in volume holography," *Optics Letters*, Vol. 19, No. 14, pp. 1079–82, July 15 1994.

[76] [HES93] L. Hesselink and M. Bashaw, "Optical memories implemented with photorefractive media," *Optical and Quantum Electronics*, Vol. 25, pp. 5611, 1993.

[77] [HEN90] P. D. Henshaw, "4D holographic optical memory," in *Proceedings of the 3–D Memory Workshop*, S. Esener and A. Craig (eds.), Snowbird, Utah, 1990.

[78] [HEU90] V. P. Heuring, "Dynamic fiber optic memories," in *Proceedings of the 3–D Memory Workshop*, S. Esener and A. Craig (eds.), Snowbird, Utah, 1990.

[79] [HUN90] S. Hunter, F. Kiamilev, S. Esener, D. Parthenopoulos, and P. Tentzepis, "Potentials of two–photon based 3–D optical memories for high performance computing," *Applied Optics*, Vol. 29, No. 14, pp. 2058–66, May 10 1990.

[80] [JAN94a] R. Jansson, H. Arwin, and I. Lundstrom, "Quasi three dimensional, n–bit optical memory based on the ellipsometric principle: model calculations," *Applied Optics*, Vol. 33, No. 29, pp. 6843–54, Oct 10 1994.

[81] [JAN94b] R. Jansson, R. Wigren, K. Jarrendakl, I. Lundstrom, and H. Arwin, "A quasi three–dimensional optical memory with n–bit memory cells based on the ellipsometric principle: concept and prototype devices," *Optics Communication*, Vol. 104, pp. 277–9, Jan 1994.

[82] [KAC90] R. Kachru, "Stimulated echo optical memory," in *Proceedings of the 3–D Memory Workshop*, S. Esener and A. Craig (eds.), Snowbird, Utah, 1990.

[83] [KAM94] A. A. Kamshilin, S. Liu, H. Tuovinen, V. V. Prokofiev, and T. Jaaskelainen, "Holographic recording of complex images by double phase conjugation in $Bi_{12}TiO_{20}$ fiber–like crystal," *Optics Letters*, Vol. 19, No. 12, pp. 907–9, June 15 1994.

[84] [KAW95b]Y. Kawata, H. Ueki, Y. Hashimoto, and S. Kawata, "Three–dimensional optical memory with a photorefractive crystal," *Applied Optics*, Vol. 34, No. 20, pp. 4105–10, July 10 1995.

[85] [LAW92] A. F. Lawrence and R. R. Birge, "The potential applications of optical phased arrays in two–photon three–dimensional optical memories," in *Proceedings of Photonic Neural Networks*, SPIE Proceedings, Vol. 1773, pp. 401–12, 1992.

[86] [LIS93] S. A. Lis, P .D. Henshaw, and M. G. Gheifetz, "Experimental demonstration of ultradense 4D holographic mass storage," *Proceedings of the SPIE*, Vol. 2026, pp. 616, 1993. Note: Invited paper.

[87] [MAN95] E. S. Maniloff, S. B. Altner, S. Bernet, F. R. Graf, A. Ren, and U. P. Wild, "Recording of 6000 holograms by use of spectral hole burning," *Applied Optics*, Vol. 34, No. 20, pp. 4140–8, July 10 1995.

[88] [MAN93] E. S. Maniloff, S. A. Altner, S. Bernet, F .R. Graf, A. Renn, and U. P. Wild, "Spectral hole burning holography in optical memory systems," *Proceedings of SPIE*, Vol. 2026, pp. 616, 1993. Note: Invited paper.

[89] [MAN91] E. S. Maniloff, and K. M. Johnson, "Maximized holographic storage," *Journal of Applied Physics*, Vol. 70, pp. 4702–7, 1991.

[90] [MAN91] E. S. Maniloff, "Dynamics of the photorefractive effect with applications to multiplexed holographic recording," Dissertation, University of Boulder, 1991.

[91] [MOK93] F. Mok, "Angle–multiplexed storage of 5000 holograms in lithium niobate," *Optical Letters*, Vol. 18, No. 11, pp. 915, 1993.

[92] [MOK92] F. Mok, D. Psaltis, and G. Burr, "Spatially and angle–multiplexed holographic random access memory," in *Proceedings of Photonic Neural Networks*, SPIE Proceedings, Vol. 1773, pp. 334–44, 1992.

[93] [PAR89] D. A. Parthenopoulous and P. M. Rentzepis, "Three–dimensional optical storage memory," *Science*, Vol. 245, pp. 843–5, Aug 25 1989.

[94] [PSA94] D. Psaltis, F. Mok, and H. S. Li, "Nonvolatile storage in photorefractive crystals," *Optics Letters*, Vol. 19, No. 3, pp. 210–2, Feb 1 1994.

[95] [PSA92] D. Psaltis, "Parallel optical memories," *Byte*, pp. 179, Sept 1992.

[96] [RHE95] U. Rhee, J. H. Caufield, C. S. Vikram, and J. Shamir, "Dynamics of hologram recording in dupont photopolymer," *Applied Optics*, Vol. 34, No. 5, pp. 846–53, Feb 10 1995.

[97] [STA72] D. L. Staebler, J. J. Anodei, and W. Phillips, "Multiple storage of thick phase holograms in $LiNbO_e$," in *7th International Quantum Electronics Conference*, Montreal, May 1972.

[98] [TAN96] T. Tanaka and S. Kawata, "Comparison of recording densities in three–dimensional optical storage systems: multilayered bit recording versus angularly multiplexed holographic recording," *Journal of the Optical Society of America A*, Vol. 13, No. 5, pp. 935–43, May 1996.

[99] [TAO95] S. Tao, Z. H. Song, D. R. Selviah, and J. E. Midwinter, "Spatioangular–multiplexing scheme for dense holographic storage", *Applied Optics*, Vol. 34, No. 29, pp. 6729–37.

[100] [WEA81] J. E. Weaver and T. K. Gaylord, "Evaluation experiments on holographic storage of binary data in electro–optic crystals," *Optical Engineering*, Vol. 20, pp. 404, 1981.

[101] [WIL85] U. P. Wild, S. E. Bucher, and f. A. Burkhalter, "Hole burning, stark effect, and data storage," *Applied Optics*, Vol. 24, pp. 1526, 1985.

[102] [YAN96] X. Yang and Z. Gu, "Three–dimensional optical data storage and retrieval system based on phase–code and space multiplexing", *Optical Engineering*, Vol. 25, No. 2, pp. 452–6, Feb 1996.

[103] [ZHA95] J. H. Zhai, Y. Ruan, and Z. G. Li, "Holographic techniques for improving the performances of frequency–domain optical storage," *Optics Communication*, Vol. 118, pp. 499–504, Aug 1995.

5.3 Error Processes and Background

[104] [ASH66] A. Ashkin, G. D. Boyd, J. M. Dziedzic, R. G. Smith, A. A. Ballman, J. J. Levinstein, and K. Nassau, "Optically–induced refractive index inhomogeneities in $LiNbO_3$ and $LiTaO_3$," *Applied Physics Letters*, Vol. 9, No. 1, July 1 1966.

[105] [BLO79] K. Blotekjaer, "Limitations on holographic storage capacity in photochromic and photorefractive media," *Applied Optics*, Vol. 18, pp. 57–67, 1979.

[106] [BAN94] P. P. Banerjee and H. Yu, "Noise analysis of two–beam coupling in a photorefractive material," *Journal of the Optical Society of America B*, Vol 11, No. 9, pp. 1809–12, Sept 1994.

[107] [BAN93] P. P. Banerjee and R. M. Misra, "Dependence of photorefractive beam fanning on beam parameters," *Optics Communications*, Vol. 100, pp. 166–72, 1993.

[108] [BAS94] M. C. Bashaw, J. F. Heanue, A. Aharoni, J. F. Walkup, and L. Hesselink, "Cross-talk considerations for angular and phase–encoded multiplexing in volume holography," *Journal of the Optical Society of America B*, Vol. 11, pp. 1820–36, 1994.

[109] [CHE69] F. S. Chen, "Optically induced change of refractive indices in $LiNbO_3$ and $LiTaO_3$," *Journal of Applied Physics*, Vol. 40, No. 8, July 1969.

[110] [CUR94b] K. Curtis and D. Psaltis, "Cross talk for angle– and wavelength–multiplexed image plane holograms," *Optics Letters*, Vol. 19, No. 21, pp. 1774–6, Nov 1994.

[111] [CUR93a] K. Curtis and D. Psaltis, "Cross talk in phase–coded holographic memories," *Journal of the Optical Society of America*, Vol. 10, No. 12, pp. 2547–50, Dec 1993.

[112] [CUR93b] K. Curtis, C. Gu, and D. Psaltis, "Cross talk in wavelength–multiplexed holographic memories", *Optics Letters*, Vol. 18, No. 12, pp. 1001–3, June 15 1993.

[113] [DET77] J. F. Detry, "Noise in charge coupled devices", Thesis at the University of Illinois at Urbana–Champaign.

[114] [FAR82] P. G. Farrell and S. J. Hopkins, "Burst–error–correcting array codes," *Radio and Electronics Engineer*, Vol. 52, No. 4, pp. 188–92, Apr 1982. ABS: Undergoes an inspection of encoding and decoding for a modified RAC+orthogonal parity to correct burst errors in a single read out situation. Their array code is suggested instead of a RS code.

[115] [FEI82] J. Feinberg, "Asymmetric self–defocusing of an optical beam from the photorefractive effect," *Journal of the Optical Society of America*, VOl. 72, No. 1, Jan 1982.

[116] [FIM94] A. Fimia, A. Belendez, and L. Carretero, "Signal–to–noise ratio of nonlinearity recorded holograms of diffuse objects," *Applied Optics*, Vol. 33, No. 32, pp. 7606–10, Nov 10, 1994.

[117] [FRE94] I. Freund, "Surface reflections and multiple scatterings in one, two, and three dimensions," *Journal of the Optical Society of America A*, Vol. 11, No. 12, pp. 3274–3283, Dec 1994.

[118] [FU95] J. Fu, J. Khoury, M. Cronin–Golomb, and C. L. Woods, "Photorefractive two–beam coupling optimal thresholding filter for additive signal–dependant noise reduction," *Applied Optics*, Vol. 34, No. 2, pp. 346–51, Jan 10 1995.

[119] [GAL94] J. T. Gallo and C. M. Verber, "Model for the effect of material shrinkage on volume holograms," *Applied Optics*, Vol. 33, No. 29, pp. 6797–804, Oct 10 1994.

[120] [GIA95] P. D. Gianino, C. L. Woods, and J. L. Horner, "Analysis of spatial light modulator contrast ratios and optical correlation," *Applied Optics*, Vol. 34, No. 29, pp. 6682–703, Oct 10 1995.

[121] [GU95a] C. Gu and F. Dai, "Cross–talk noise reduction in volume holographic storage with an extended recording reference," *Optics Letters*, Vol. 20, No. 22, pp. 2336–8, Nov 15, 1995.

[122] [GU92] C. Gu, J. Hong, I. McMichael, R. Saxena, and F. Mok, "Cross–talk–limited storage capacity of volume holographic memory," *Journal of the Optical Society of America A*, Vol. 9, No. 11, pp. 1978–83, Nov 1992.

[123] [GUO96] R. Guo and Z. Ling, "Diffraction efficiency of photorefractive index gratings,", in *Optics Communications*, Vol. 125, pp. 101–4, April 1 1996.

[124] [HE94] Q. B. He and P. Yeh, "Fanning noise reduction in photorefractive amplifiers using incoherent erasures," *Applied Optics*, Vol. 33, No. 2, pp. 283, Jan 10 1994.

[125] [HO95] C. Ho, "Calculating the performance of optical communication systems with modal noise by saddlepoint method," *Journal of Lightwave Technology*, Vol. 13, No. 9, pp. 1821–5, Sept 1995.

[126] [HON90] J. Hong, P. Yeh, D. Psaltis, and D. Brady, "Diffraction efficiency of strong volume holograms," *Optics Letters*, Vol. 15, pp 344–346, 1990.

[127] [HOR92] M. Horowitz, R. Daisy, and B. Fischer, "Signal–to–pump ratio dependence of buildup and decay rates in photorefractive nonlinear two–beam coupling," *Journal of the Optical Society of America B*, Vol. 9, No. 9, Sept 1992.

[128] [HUI76] J. P. Huignard, F. Micheron, and E. Spitg, "Optical properties of solids," North Holland, Amsterdam, 1976, Chapter 16.

[129] [INS89] INSPEC eds, "Properties of lithium niobte", EMIS Datareview Series No. 5, INSPEC: London, 1989

[130] [KAM95] K. Kamra and K. Singh, "Characterization of beam fanning in $BaTiO_3$ under biasing illumination and its application as log processor," *Optical Engineering*, Vol. 34, No. 8, Aug 1995.

[131] [KAW95a] W. Kawakami and K. I. Kitayama, "Optical error–correction coding encoder and decoder: design considerations," *Applied Optics*, Vol. 34, No. 23, pp. 5064–73, Aug 10 1995.

[132] [LEE88] H. Lee, "Cross–talk effects in multiplexed volume holograms," *Optics Letters*, Vol. 13, No. 10, pp. 874–6, Oct 88.

[133] [LI96] H. S. Li and J. Hong, "Nonuniformity in hologram diffraction efficiency from time–constant error in the recording schedule," *Journal of the Optical Society of America B*, Vol. 13, No. 5, pp. 894–9, May 1996.

[134] [LI95] H. S. Li and D. Psaltis, "Alignment sensitivity of holographic three–dimensional disks," *Journal of the Optical Society of America A*, Vol. 12, No. 9, pp. 1902–12, Sept 1995.

[135] [LIU94] J. Liu, P. P. Banerjee, and Q. W. Song, "Role of diffusive, photovoltaic, and thermal effects in beam fanning in $LiNbO_3$," *Journal of the Optical Society of America B*, Vol. 11, No. 9 pp. 1688–93, Sept 1994.

[136] [LOH96] A. W. Lohmann, R. G. Dorsch, D. Mendlovic, Z. Zalevsky, and C. Ferreira, "Space–bandwidth product of optical signals and systems," *Journal of the Optical Society of America A*, Vol. 13, No. 3, pp. 470–3, Mar 1996.

[137] [MIR94] S. V. Miridonov, A. V. Khomenko, D. Tentori, and A. A. Kanishilin, "Information capacity of holograms in photorefractive crystals," *Optics Letters*, Vol. 19, No. 7, pp. 502–4, Apr 1 1994.

[138] [MIT94a] P. A. Mitkas and L. J. Irakliotis, "Three–dimensional optical storage for database processing," invited paper, *Journal of Optical Memories and Neural Networks*, Vol. 3, No. 2, pp. 217–29, 1994.

[139] [MIT94b] P. A. Mitkas, L. J. Irakliotis, R. R. Beyette Jr., S. A. Feld, and C. W. Wilmsen, "An optoelectronic data filter for selection and projection," *Applied Optics*, Vol. 33, No. 8, pp. 1345–53. March 10 1994.

[140] [MOH76] M. G. Moharam and L. Young, "Hologram writing by the photorefractive effect with gaussian beams at constant applied voltage," *Journal of Applied Physics*, Vol. 47, No. 9, Sept 1976.

[141] [MON95] G. Montemezzani, A. A. Zozulya, L. Czaia, and D. Z. Anderson, "Origin of the lobe structure in photorefractive beam fanning," *Physical Review A*, Vol. 52, No. 2, Aug 1995.

[142] [NEI95b] M. A. Neifeld, "Improvement in the capacity of computer generated holographic storage using the lee method with sparse multivalued reconstructions," *Applied Optics*, Vol. 34, No. 8, pp. 1396–1400, Mar 10 1995.

[143] [NEI93] M. A. Neifeld, "Computer–generated holography for optical memory using sparse data words: capacity and error tolerance," *Applied Optics*, Vol. 32, No. 26, pp. 5125–34, Sept 10 1993.

[144] [NOR94] G. P. Nordin and P. Asthana, "Achieving a minimum signal–to–noise ratio in angularly multiplexed volume holographic optical data storage systems," *SPIE*, Vol. 2297, pp. 392–401, 1994.

[145] [NOR93] G. P. Nordin and P. Asthana, "Effects of cross talk on fidelity in page–oriented volume holographic optical data storage," *Optical Letters*, Vol. 18, No. 18, pp. 1553–5, Sept 15 1993.

[146] [PEP90] D. M. Pepper, J. Feinberg, and N. V. Kukhtarev, "The photorefractive effect," *Scientific American*, pp. 62–74, Oct 1990.

[147] [PRO90] A. M. Prokhorov and Y. S. Kuz'minov, "Physics and chemistry of crystalline lithium niobate," trans. by T. M. Pyankova and O. A. Zilbert, IOP Publishing: Brystol England, 1990.

[148] [RAS93] L. Rastani, "Storage capacity and cross talk in angularly multiplexed holograms: two case studies," *Applied Optics*, Vol. 32, No. 20, pp. 3772–7, July 10 1993.

[149] [SAL91] B. E. A. Saleh and M. C. Teich, *Fundamentals of Photonics*, John Wiley & Sons: New York, 1991.

[150] [SEG93] M. Segev, D. Engin, A. Yariv, and G. C. Valley, "Temporal evolution of fanning in photorefractive materials," *Optics Letters*, Vol. 18, No. 12, June 15, 1993.

[151] [SEN94] H. Senagalg and P. Saari, "Diffraction efficiency in space–and–time–domain holography," *Journal of the Optical Society of America B*, Vol. 11, No. 2, pp. 372–9, Feb 1994.

[152] [SNY95] D. L. Snyder, C. W. Heistrom, A. D. Lanterman, M. Faisal, and R. L. White, "Compensation for readout noise in CCD images," *Journal of the Optical Society of America A*, Vol. 12, No. 2, pp. 272–83, Feb 1995.

[153] [SNY93] D. L. Snyder, A. M. Hammond, and R. L. White, "Image recovery form data acquired with a charge–coupled–device camera," *Journal of the Optical Society of America A*, Vol. 10, pp. 1014–23, 1993.

[154] [STR94] R. W. Stroud and W. T. Rhodes, "Comparison of diffraction efficiencies for single–exposure and unity–contrast multiple–exposure holograms," *Applied Optics*, Vol. 33, No. 17, pp. 3627–9, June 10 1994.

[155] [TSU94] T. Tsujioka, Y Shimizu, and M. Irie, "Cross talk in photon–mode photochromic multi–wavelength recording," *Applied Physics*, Vol. 33, No. 4A, pp. 1914–9, Apr 1994.

[156] [VAL83] G. C. Valley and M. B. Klein, "Optimal properties of photorefractive materials for optical data processing," *Optical Engineering*, Vol. 22, No. 6, pp. 704–11, Nov 1983.

[157] [WEI85] R. S. Weis and T. K. Gaylord, "Lithium niobate: summary of physical properties and crystal structure," *Applied Physics A (Solids and Surfaces)*, Vol. 37, pp. 191–203, 1985.

[158] [WUL94] J. R. Wullert II and Y. Lu, "Limits of the capacity and density of holographic storage," *Applied Optics*, Vol. 33, No. 11, pp. 2192–198, Apr 10 1994.

[159] [YAR95] A. Yariv, S. Orlov, G. Rakuljee, and V. Leyva, "Holographic fixing, readout, and storage dynamics in photorefractive materials," *Optics Letters*, Vol. 20. No. 11, pp. 1334–7, June 1 1995.

[160] [YI95a] X. Yi, P. Yeh, and C. Gu, "Cross–talk noise in volume holographic memory with spherical reference beams," *Optics Letters*, Vol. 20, No. 17, pp. 1812–4, Sept 1 1995.

[161] [YI95b] X. Yi, S. Campbell, P. Yeh, and C. Gu, "Statistical Analysis of Cross–Talk, Noise, and Storage Capacity in Volume Holographic Memories: Image Plane Holograms," *Optics Letters*, Vol. 20, No. 7, pp. 779–81, Apr 1 1995.

[162] [YI94] X. Yi, P. Yeh, and C. Gu, "Statistical analysis of cross–talk noise and storage in volume holographic memory," *Optical Letters*, Vol. 19, No. 19, pp. 1580–2, Oct 1 1994.

[163] [ZHO95] H. Zhou, F. Zhaw, and F. T. S. Yu, "Angle–dependant diffraction efficiency in a thick photorefractive hologram," *Applied Optics*, Vol. 34, No. 8, pp. 1303–9, Mar 10 1995.

☆U.S. GOVERNMENT PRINTING OFFICE: 1997-509-127-61023

MISSION OF ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Material Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.